An Object Template Approach to Manipulation for Semi-autonomous Avatar Robots

TECHNISCHE UNIVERSITÄT DARMSTADT

Vom Fachbereich Informatik der
Technischen Universität Darmstadt
zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte

Dissertation

von

M.Sc. Alberto Isay Romay Tovar
(geboren in Mexiko-Stadt)

Referent:	Prof. Dr. Oskar von Stryk
Koreferent:	Prof. Dr. Tamim Asfour
	(Karlsruher Institut für Technologie, Deutschland)
Tag der Einreichung:	17.02.2016
Tag der mündlichen Prüfung:	30.03.2016

D17
Darmstadt 2017

Para mi "Pa", la "Vivi", y el "Nick".

Acknowledgements

This thesis was created during my time as doctor candidate at the *Simulation, Systems Optimization and Robotics Group (SIM)* at the *Department of Computer Science of the Technische Universität Darmstadt*.

I thank for the support of all my colleagues at SIM group which are now also my friends. They were the best „*Integrationskurs*" I could ever had.

I want thank Christine Blümler for her kind support and motivation while writing this work.

I want to specially thank my supervisor Prof. Dr. rer. nat. Oskar von Stryk for his guidance, support, and kindness. He opened his lab doors to me giving me the opportunity to experience, literally, the most advanced technology in robotics.

Likewise, I want to thank Prof. Dr. Tamim Asfour for becoming the second referee of my thesis. I am grateful for his interest in my work and for his supportive comments.

I thank all members of Team ViGIR, specially David Conner for sharing his experiences with the DARPA Robotics Challenge project.

Contents

Abstract **1**

Zusammenfassung **3**

1 Introduction **7**
1.1 Context . . . 9
1.2 Contribution and Contents . . . 10

2 Background **13**
2.1 Remote Mobile Manipulation Robots . . . 14
2.1.1 Background from Pure Teleoperated Manipulation Robots . . . 14
2.1.2 Background from Full Autonomous Manipulation Robots . . . 14
2.1.3 Background from Semi-autonomous Manipulation Robots . . . 16
2.2 Object Manipulation Theory . . . 17
2.2.1 The Concept of Affordances . . . 17
2.2.2 Affordance-based Applications in Robotics . . . 18
2.2.3 Discussion . . . 19

3 The Concept of Object Templates **21**
3.1 Related Work . . . 21
3.2 Contribution . . . 22
3.3 Definition of an Object Template . . . 23
3.4 Information Contained in an Object Template . . . 24
3.4.1 Physical Information . . . 24
3.4.2 Abstract Information . . . 25

4 Object Template Framework for Remote Manipulation Control **29**
4.1 Related Work . . . 29
4.2 Contribution . . . 30
4.3 Implementation of the Concept of Object Templates . . . 30
4.3.1 Object Template Implementation . . . 31
4.3.2 Grasp Template Implementation . . . 33
4.3.3 Stand Template Implementation . . . 35
4.4 Affordances in an Object Template . . . 37
4.5 The Concept of Usabilities in an Object Template . . . 37
4.5.1 Definition . . . 38
4.5.2 Implementation . . . 39
4.6 Manipulation Implementation in an Object Template . . . 41
4.6.1 Constrained Motion Planning . . . 43
4.6.2 End-effectors and Grasp Agnostic Manipulation . . . 43
4.6.3 Planning with Grasped Objects . . . 44

4.7 Allowing Versatility and Robustness to Solve Manipulation Tasks 44
4.7.1 Object Classification 45
4.7.2 Manipulation Classification 46
4.7.3 Manipulation Transferring Types 47

5 Human-Robot Interaction Through Object Templates 49
5.1 Related Work 49
5.2 Contribution 50
5.3 Operator Control Station 50
5.3.1 Requirements 50
5.3.2 World Model Generation through Robots Sensors 51
5.4 Integration of Object Templates into an Operator Control Station 52
5.4.1 Communication Constraints 53
5.4.2 Object Template Server 54
5.5 General Manipulation Task Pipeline 54
5.6 A Multi-operator Approach 56
5.6.1 Supervisor Operator (Planning) 57
5.6.2 Primary Operator (Action) 57
5.6.3 Auxiliary Operator (Perception) 57
5.6.4 Immersed Operator (Observation) 58
5.7 Collaborative Autonomy 58

6 Experiments, Applications, and Results 61
6.1 Robot Hardware Used for Experimentation 61
6.1.1 Atlas 61
6.1.2 THORMANG 63
6.1.3 Different End-effectors and Hands 63
6.2 DARPA Robotics Challenge 64
6.2.1 VRC 65
6.2.2 DRC Trials 66
6.2.3 DRC Finals 72
6.3 Laboratory Experimentation 76
6.3.1 Timeline Results Analysis 79
6.3.2 Experiments for Manipulation Skills Transferring 81
6.3.3 Experiments with Object Template Usabilities 83
6.4 Experiments from other Research Group 86

7 Conclusions 89
7.1 Summary of the Contributions 89
7.2 Outlook 91

Bibliography 93

Own Publications 103

List of Figures

1.1 Human operator and remote semi-autonomous robot 8
1.2 Object template approach chapters graphic 9

2.1 Teleoperation robots 15
2.2 Full autonomous robots. 16
2.3 Semi-autonomous robots. 17

3.1 Drill: DeWalt DCD980M2. 25
3.2 External physical information of an object. 26

4.1 Relationship between objects, grasps and stand poses libraries. 31
4.2 Cutting tool, high detail mesh, and low detail mesh. 33
4.3 Example of an object template. 33
4.4 Using a “ghost hand” the final grasps can be visualized. 34
4.5 Using a “ghost robot” the view pre-grasp and final-grasp poses. 35
4.6 Two different stand poses to open a door. 36
4.7 Object usabilities: the drill template and the paint roller template. 39
4.8 Robot’s end-effector path for cutting a circle in the wall. 40
4.9 Frame of reference of the arm joints with a “drill dit”. 41
4.10 Transformations to generate the correct path to cut a circle. 42
4.11 Circular path plan to turn a valve 360 degrees. 43
4.12 Floating objects. 46
4.13 Constrained objects. 47
4.14 Manipulation classification. 48

5.1 Operator control station. 51
5.2 World model of a valve scenario. 52
5.3 Interactive marker. 52
5.4 Manipulation widgets for each hand. 53
5.5 Description of the functions in the manipulation widget. 54
5.6 Object template server. 55
5.7 Workflow of a manipulation task using object templates. 56
5.8 Multi-operator approach. 57

6.1 Humanoid robots used 62
6.2 Different end-effectors and hands 64
6.3 VRC manipulation rask 66
6.4 Manipulation tasks in the DRC Trials 67
6.5 Request to walk to the hose. 69
6.6 Request to grasp hose. 69
6.7 Request to walk away from reel. 70
6.8 Request to walk to the wye. 70
6.9 Robot commanded to bring the hand near the wye and then touch it. 71

6.10 Attach hose and turn nozzle (teleoperated). 71
6.11 DRC external footage from Team Hector at the door task. 74
6.12 Robot's camera view from the door task at the DRC. 75
6.13 Robot's camera view from Valve task at the DRC. 75
6.14 Robot's camera view from door task. 77
6.15 Robot's camera view from valve task. 78
6.16 "Turn valve" behavior execution in `High` autonomy level 78
6.17 Timeline of actions during DRC Finals tasks and laboratory experimentation. . . . 80
6.18 The robot generating a clock-wise circular path to draw a circle 82
6.19 The robot generating a circular path to turn the steering wheel. 82
6.20 The robot using the "turning" affordance of a valve template. 83
6.21 Johnny drawing circles. 84
6.22 Atlas drawing a circle with a drill. 85
6.23 An unreachable valve. 86
6.24 Turning an unreachable valve with usabilities. 87
6.25 Team Valor at DRC Finals 87
6.26 Team Valor opening a door 88
6.27 Team Valor pulling down an electric break 88

List of Listings

4.1 Short version of the object template for the DRC Drill. 32
4.2 Grasp template for a right robotic hand with two fingers. 35
4.3 A short version of the stand template for a door. 37

List of Tables

3.1 Physical Information of the object. 25

4.1 Affordances described in object templates. 38
4.2 Manipulation classes. 47

6.1 Comparison of Humanoid Systems 63
6.2 Different end-effectors and hands 65
6.3 Detailed Team ViGIR VRC scores 66
6.4 Top six teams VRC results 67
6.5 Hose Task timetable for all Atlas teams. 72
6.6 Time table of DRC tasks and laboratory evaluation. 81

Abstract

Nowadays, the first steps towards the use of mobile robots to perform manipulation tasks in remote environments have been made possible. This opens new possibilities for research and development, since robots can help humans to perform tasks in many scenarios. A remote robot can be used as avatar in applications such as for medical or industrial use, in rescue and disaster recovery tasks which might be hazardous environments for human beings to enter, as well as for more distant scenarios like planetary explorations. Among the most typical applications in recent years, research towards the deployment of robots to mitigate disaster scenarios has been of great interest in the robotics field.

Disaster scenarios present challenges that need to be tackled. Their unstructured nature makes them difficult to predict and even though some assumptions can be made for human-designed scenarios, there is no certainty on the expected conditions. Communications with a robot inside these scenarios might also be challenged; wired communications limit reachability and wireless communications are limited by bandwidth. Despite the great progress in the robotics research field, these difficulties have prevented the current autonomous robotic approaches to perform efficiently in unstructured remote scenarios. On one side, acquiring physical and abstract information from unknown objects in a full autonomous way in uncontrolled environmental conditions is still an unsolved problem. Several challenges have to be overcome such as object recognition, grasp planning, manipulation, and mission planning among others. On the other side, purely teleoperated robots require a reliable communication link robust to reachability, bandwidth, and latency which can provide all the necessary feedback that a human operator needs in order to achieve sufficiently good situational awareness, e.g., worldmodel, robot state, forces, and torques exerted. Processing this amount of information plus the necessary training to perform joint motions with the robot represent a high mental workload for the operator which results in very low execution times. Additionally, a pure teleoperated approach is error-prone given that the success in a manipulation task strongly depends on the ability and expertise of the human operating the robot. Both, autonomous and teleoperated robotic approaches have pros and cons, for this reason a middle ground approach has emerged.

In an approach where a human supervises a semi-autonomous remote robot, strengths from both, full autonomous and purely teleoperated approaches can be combined while at the same time their weaknesses can be tackled. A remote manipulation task can be divided into sub-tasks such as *planning*, *perception*, *action*, and *evaluation*. A proper distribution of these sub-tasks between the human operator and the remote robot can increase the efficiency and potential of success in a manipulation task. On the one hand, a human operator can trivially plan a task (planning), identify objects in the sensor data acquired by the robot (perception), and verify the completion of a task (evaluation). On the other hand, it is challenging to remotely control in joint space a robotic system like a humanoid robot that can easily have over 25 Degrees of Freedom (DOF). For this reason, in this approach the complex sub-tasks such as motion planning, motion execution, and obstacle avoidance (action) are performed autonomously by the remote robot. With this distribution of tasks, the challenge of converting the operator intent into a robot action arises.

This thesis investigates concepts of how to efficiently provide a remote robot with the operator intent in a flexible means of interaction. While current approaches focus on an object-grasp-centered means of interaction, this thesis aims at providing physical and abstract properties of the objects of interest. With this information, the robot can perform autonomous sub-tasks like locomotion through the environment, grasping objects, and manipulating them at an affordance-level avoiding collisions with the environment in order to efficiently accomplish the manipulation task needed.

For this purpose, the concept of Object Template (OT) has been developed in this thesis. An OT is a virtual representation of an object of interest that contains information that a remote robot can use to manipulate such object or other similar objects. The object template concept presented here goes beyond state-of-the-art related concepts by extending the robot capabilities to use affordance information of the object. This concept includes physical information (mass, center of mass, inertia tensor) as well as abstract information (potential grasps, affordances, and usabilities). Because humans are very good at analysing a situation, planning new ways of how to solve a task, even using objects for different purposes, it is important to allow communicating the *planning* and *perception* performed by the operator such that the robot can execute the *action* based on the information contained in the OT. This leverages human intelligence with robot capabilities. For example, as an implementation in a 3D environment, an OT can be visualized as a 3D geometry mesh that simulates an object of interest. A human operator can manipulate the OT and move it so that it overlaps with the visualized sensor data of the real object. Information of the object template type and its pose can be compressed and sent using low bandwidth communication. Then, the remote robot can use the information of the OT to approach, grasp, and manipulate the real object.

The use of remote humanoid robots as avatars is expected to be intuitive to operators (or potential human response forces) since the kinematic chains and degrees of freedom are similar to humans. This allows operators to visualize themselves in the remote environment and think how to solve a task, however, task requirements such as special tools might not be found. For this reason, a flexible means of interaction that can account for allowing improvisation from the operator is also needed. In this approach, improvisation is described as "*a change of a plan on how to achieve a certain task, depending on the current situation*". A human operator can then improvise by adapting the affordances of known objects into new unknown objects. For example, by utilizing the affordances defined in an OT on a new object that has similar physical properties or which manipulation skills belong to the same class.

The experimental results presented in this thesis validate the proposed approach by demonstrating the successful achievement of several manipulation tasks using object templates. Systematic laboratory experimentation has been performed to evaluate the individual aspects of this approach. The performance of the approach has been tested in three different humanoid robotic systems (one of these robots belongs to another research laboratory). These three robotic platforms also participated in the renowned international competition DARPA Robotics Challenge (DRC) which between 2012 and 2015 was considered the most ambitious and challenging robotic competition.

Keywords: *remote robots, manipulation control, supervised semi-autonomous robots, object manipulation, affordances*.

Zusammenfassung

Die ersten Schritte zur Verwendung von mobilen Robotern zur Durchführung von Manipulationsaufgaben in entfernten Umgebungen sind in jüngster Zeit möglich gemacht worden. Dies eröffnet neue Möglichkeiten für Forschung und Entwicklung, da Roboter dem Menschen helfen können, Aufgaben in vielen Szenarien durchzuführen. Ein entfernter Roboter kann als Avatar in Anwendungen, wie zum Beispiel für medizinische oder industrielle Nutzung, in Rettungsaufgaben, die gefährliche Umgebungen für Menschen darstellen können, sowie für weiter entfernte Szenarien wie planetare Exploration verwendet werden. Bei den typischsten Anwendungen der letzten Jahre war die Forschung zum Einsatz von Robotern zur Minderung von Katastrophenszenarien von großem Interesse im Bereich der Robotik.

Katastrophenszenarien stellen Herausforderungen dar, die man in Angriff nehmen muss. Ihre unstrukturierte Natur macht es schwierig, sie vorherzusagen, und obwohl einige Annahmen für menschlich gestaltete Szenarien gemacht werden können, gibt es keine Gewissheit über die erwartbaren Bedingungen. Kommunikation mit einem Roboter in diesen Szenarien könnte ebenfalls problematisch sein; Drahtgebundene Kommunikation begrenzt Erreichbarkeit und drahtlose Kommunikation ist durch Bandbreite begrenzt. Trotz der großen Fortschritte in der Robotik-Forschung limitieren diese Schwierigkeiten die derzeitigen Ansätze für autonomen Roboter, um effizient in unstrukturierten entfernten Szenarien zu funktionieren. Auf der einen Seite ist es immer noch ein ungelöstes Problem, physikalische und abstrakte Informationen von unbekannten Objekten autonom in unkontrollierten Umgebungsbedingungen zu erlangen. Mehrere Herausforderungen müssen überwunden werden, wie unter anderem die Objekterkennung, die Planung, die Manipulation und die Missionsplanung. Auf der anderen Seite erfordern reine teleoperierte Roboter eine zuverlässige Kommunikationsverbindung, die eine robuste Erreichbarkeit, Bandbreite und Latenz hat und alle notwendigen Rückmeldungen liefern kann, die ein menschlicher Bediener benötigt, um ein ausreichend gutes Situationsbewusstsein zu erreichen, z. B. Weltmodell, Roboterzustand, Kräfte und wirkende Drehmomente. Die Verarbeitung dieser Informationsmenge und die notwendige Schulung zur Durchführung von Gelenkbewegungen mit dem Roboter stellen eine hohe geistige Arbeitsbelastung für den Bediener dar, die zu sehr geringen Ausführungszeiten führt. Zusätzlich ist ein reiner teleoperierter Ansatz fehleranfällig, da der Erfolg in einer Manipulationsaufgabe stark von der Fähigkeit und dem Fachwissen des menschlichen Bedieners des Roboters abhängt. Sowohl autonome als auch teleoperierte Roboteransätze haben Vor- und Nachteile. Aus diesem Grund ist ein Ansatz entwickelt worden, der beide Welten miteinander vereint.

In einem Ansatz, bei dem ein Mensch einen semi-autonomen Fernroboter überwacht, können die Stärken von beiden, vollen autonomen und rein teleoperierten Ansätzen kombiniert werden, während gleichzeitig ihre Schwächen angegangen werden können. Eine Fernmanipulationsaufgabe kann in Unteraufgaben wie Planung, Wahrnehmung, Aktion und Auswertung unterteilt werden. Eine ordnungsgemäße Verteilung dieser Unteraufgaben zwischen dem menschlichen Bediener und dem entfernten Roboter kann die Effizienz und das Potenzial des Erfolgs in einer Manipulationsaufgabe erhöhen. Ein Mensch kann einerseits eine Aufgabe planen (Planung), Objekte in den vom Roboter erfassten Sensordaten identifizieren (Wahrnehmung), und den Abschluss einer Aufgabe (Auswertung) validieren. Auf der anderen Seite ist es anspruchsvoll, im Gelenkraum ein Robotersystem wie einen humanoiden Roboter fernsteuern zu können, der

leicht über 25 Freiheitsgrade haben kann. Aus diesem Grund werden bei diesem Ansatz die komplexen Teilaufgaben wie Bewegungsplanung, Bewegungsausführung und Hindernisvermeidung (Aktion) autonom vom entfernten Roboter durchgeführt. Mit dieser Aufteilung der Aufgaben entsteht die Herausforderung, die Absicht des Betreibers in eine Aktion des Roboters umzuwandeln.

Diese Arbeit untersucht Konzepte, wie eine flexible Interaktion zwischen der Intension des Bedieners und einen entfernten Roboter gestaltet werden kann. Während sich aktuelle Ansätze auf objektgreifzentrierte Ansätze konzentrieren, zielt diese Arbeit auf die Bereitstellung physikalischer und abstrakter Eigenschaften der betrachteten Objekte ab. Mit diesen Informationen kann der Roboter autonome Teilaufgaben durchführen, wie die Fortbewegung durch die Umgebung, das Greifen von Objekten und die Manipulation auf einem Affordanz-Level, wodurch Kollisionen mit der Umgebung vermieden werden, um die erforderliche Manipulationsaufgabe effizient beenden zu können.

Zu diesem Zweck wurde das Konzept der Objekt-Vorlage (OV) in dieser Arbeit entwickelt. Eine OV ist eine virtuelle Darstellung eines Objekts von Interesse, die Informationen enthält, die ein entfernter Roboter verwenden kann, um ein solches Objekt oder andere ähnliche Objekte zu manipulieren. Das hier vorgestellte OV-Konzept geht über aktuell verwendete Konzepte hinaus, indem es die Fähigkeiten des Roboters erweitert, um Affordanz-Informationen des Objekts zu nutzen. Dieses Konzept beinhaltet physikalische Informationen (Masse, Schwerpunkt, Trägheitstensor) sowie abstrakte Informationen (potentielle Griffe, Affordanzen und Nutzen). Da Menschen sehr gut darin sind, eine Situation zu analysieren und neue Wege zur Lösung von Aufgaben zu finden, sogar mit Objekten für unterschiedliche Zwecke, ist es wichtig, die Planung und die Wahrnehmung, die der Bediener durchführt, zu kommunizieren. So kann der Roboter die Aktion basierend auf den in der OV enthaltenen Informationen ausführen. Dies vereint die menschliche Intelligenz mit Roboterfähigkeiten. Beispielsweise kann als eine Implementierung in einer 3D-Umgebung eine OV als ein 3D-Geometriegitter visualisiert werden, das ein Objekt von Interesse simuliert. Ein Mensch kann die OV manipulieren und verschieben, so dass sie mit den visualisierten Sensordaten des realen Objekts überlappt. Informationen des OV-Typs und seiner Pose können komprimiert und gesendet werden, indem Kommunikation mit geringer Bandbreite verwendet wird. Dann kann der entfernte Roboter die Information der OV verwenden, um das reale Objekt anzunähern, zu erfassen und zu manipulieren.

Die Verwendung von entfernten humanoiden Robotern als Avatare wird voraussichtlich intuitiv für Operatoren (oder potentielle menschliche Rettungskräfte) sein, da die kinematischen Ketten und Freiheitsgrade dem Menschen ähnlich sind. Dies ermöglicht es Bedienern, sich in der entfernten Umgebung zu orientieren und eine Lösung für eine Aufgabe zu finden, wobei die notwendigen Aufgabenanforderungen, wie spezielle Werkzeuge, dann eventuell nicht gefunden werden können. Aus diesem Grund ist auch ein flexibles Interaktionsverfahren erforderlich, das die Improvisation des Betreibers berücksichtigen kann. In diesem Ansatz wird die Improvisation als "eine änderung eines Plans zur Erreichung einer bestimmten Aufgabe, abhängig von der aktuellen Situation" beschrieben. Eine menschliche Bedienperson kann dann improvisieren, indem sie die Vorteile von bekannten Objekten an neue unbekannte Objekte anpasst. Beispielsweise können durch Verwendung der in einer OV definierten Vorteile auf einem neuen Objekt, das ähnliche physikalische Eigenschaften aufweist oder dessen Manipulationsfähigkeiten derselben Klasse angehören, verwendet werden.

Die in dieser Arbeit vorgestellten experimentellen Ergebnisse bestätigen den vorgeschlagenen Ansatz, indem sie das erfolgreiche Erreichen mehrerer Manipulationsaufgaben mit Objekt-

Vorlagen demonstrieren. Systematische Laborexperimente wurden durchgeführt, um die einzelnen Aspekte dieses Ansatzes zu bewerten. Die Leistung des Ansatzes wurde in drei verschiedenen humanoiden Robotersystemen getestet (einer dieser Roboter gehört einem anderen Forschungslabor). Diese drei Roboterplattformen nahmen auch am renommierten internationalen Wettbewerb DARPA Robotics Challenge (DRC) teil, der zwischen 2012 und 2015 als der ehrgeizigste und herausforderndste Roboterwettbewerb angesehen wurde.

1 Introduction

Using robots to perform remote manipulation tasks, such as for disaster relief or space exploration, has become of great interest in the research community during the last years. Unstructured and potentially degraded environments that might be hazardous for humans to explore also present challenges for robot systems that need to be tackled. Environment conditions such as lighting, obstacles, and manipulation of objects are often easy for humans to handle, however, these conditions increase the difficulty for robots to identify objects of interest, perform self-localization, and manipulate objects which might also be unknown.

Executing robotic tasks in remote environment has been mainly approached from two different perspectives. On the one hand, full autonomous manipulation using humanoid robots under unconstrained conditions is still an unsolved problem. On the other hand, the use of purely teleoperated robots requires dealing with communication constraints such as reachability, loss of information, and latency. Also, performing teleoperated manipulation tasks is error-prone given the high mental workload and the lack of situational awareness from the human operator.

For fully autonomous robots to efficiently navigate and interact in remote unconstrained environments, the capabilities are required to include robust perception systems, extensive databases with information from the objects of interest, efficient online grasping algorithms and the ability to improvise upon unforeseen situations. These capabilities have a high degree of complexity and autonomous robust solutions are nowadays not feasible. For purely teleoperated robots, the capabilities required include near real-time communications to provide proper feedback and situational awareness to the operator. Additionally, for remote teleoperated robots, the ability to perform manipulation tasks strongly depends on the expertise of the human operating the robot.

Taking a middle ground perspective, a human-supervised semi-autonomous robot is an approach that can leverage strengths from both, fully autonomous and purely teleoperated approaches, while at the same time tackling their weaknesses. In this approach, the robot is intended to be used as a remote manifestation of a human, or so-called avatar [71]. An avatar robot should be enabled to perform the operator's intent while at the same time it should be able to perform autonomous low-level tasks. For example, robots are good at processing large amounts of data and perform complex calculations (for motion planning and locomotion) and humans have great abilities such as perception, planning, and improvisation, thus, it make sense to complement each other's abilities to perform manipulation tasks in complex environments.

Based on the hierarchical robot paradigm primitives Sense, Plan, Act (SPA) [63] in a manipulation task four main sub-tasks can be identified:

1. *Sense:* Sensor data acquisition to provide situational awareness as well as for object of interest recognition and localization.

2. *Plan:* Task objective identification and strategy formulation to achieve the task goal.

3. *Act:* Joint motion generation for locomotion, grasping, and manipulation considering motion planning, motion execution, and obstacle avoidance.

4. *Evaluation:* Analysis of the environment changes to verify task goal achievement.

From the perspective of fully autonomous robots, all these sub-tasks should be performed by the robot. From the perspective of pure teleoperated robots, all these sub-tasks should be performed by the operator. Finally, from the perspective of a semi-autonomous avatar robot, these sub-tasks can be performed by the robot and the operator according to their corresponding strengths and weakness. Figure 1.1 shows an example of division of these sub-tasks.

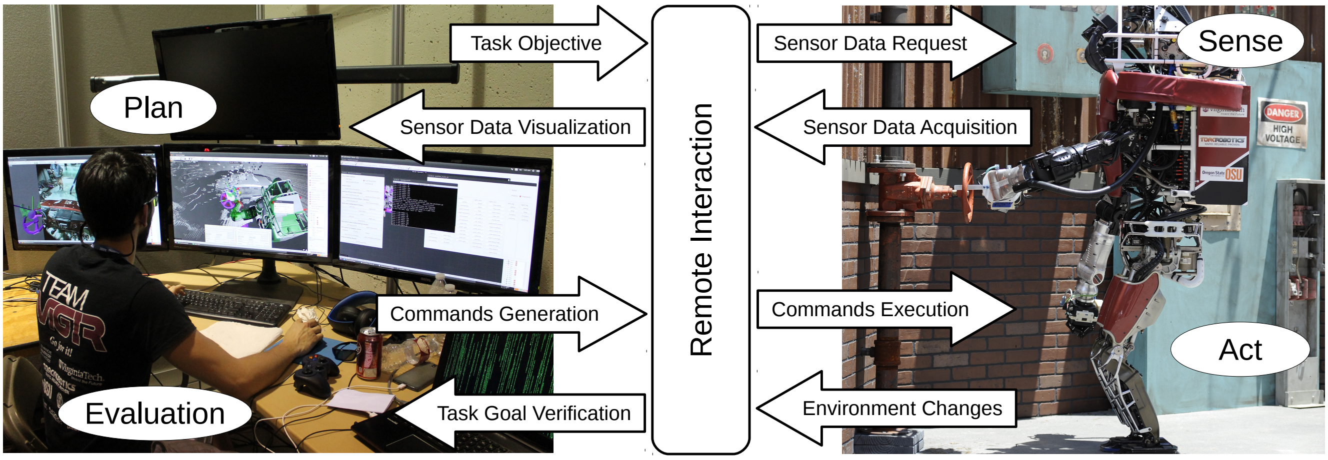


Figure 1.1: Human operator (left) supervising and generating high-level task commands to be executed by the remote semi-autonomous robot (right).

To properly divide this sub-tasks between an operator and the remote robot, an interaction method is required. This interaction method should enable a human operator to perceive and gain situational awareness of the remote environment while at the same time provide the robot with the ability to receive information from the objects of interest and autonomously execute commands from the operator. To reduce the amount of mental workload, the operator should be able to provide high-level task-objective information while low-level tasks such as motion planning, motion execution, and obstacle avoidance are still autonomously performed by the robot.

Enabling command communication between a human operator and a remote semi-autonomous robot requires the ability to convert the human operator intention into a robot action. For this reason, it is important to have a source of information that can be human-friendly and robot-friendly at the same time. For humans it is straightforward to understand the environment and make sense of the objects and objectives that a task implies. If this understanding can be abstracted and described in a way that a robotic system can use it in a general form, then it is possible to transfer the human operator intent into a robotic action. Figure 1.2 shows a graphic that describes how the concept of an object (describing its physical and abstract properties) can be abstracted and represented through an *object template*, which is a central concept developed in this thesis. Object templates representing objects in the real world can be organized in a system that provides a human operator with the ability to transmit intent and at the same time an avatar semi-autonomous robot can use the information represented in this object templates to perform manipulation (including locomotion planning if required) of real objects.

Object templates can also be used in a collaborative autonomy approach. In collaborative autonomy, object template information is abstracted into a high-level behavior. This high-level behavior can coordinate tasks between the human supervisor and the avatar robot so that remote manipulation tasks can be systematically executed.

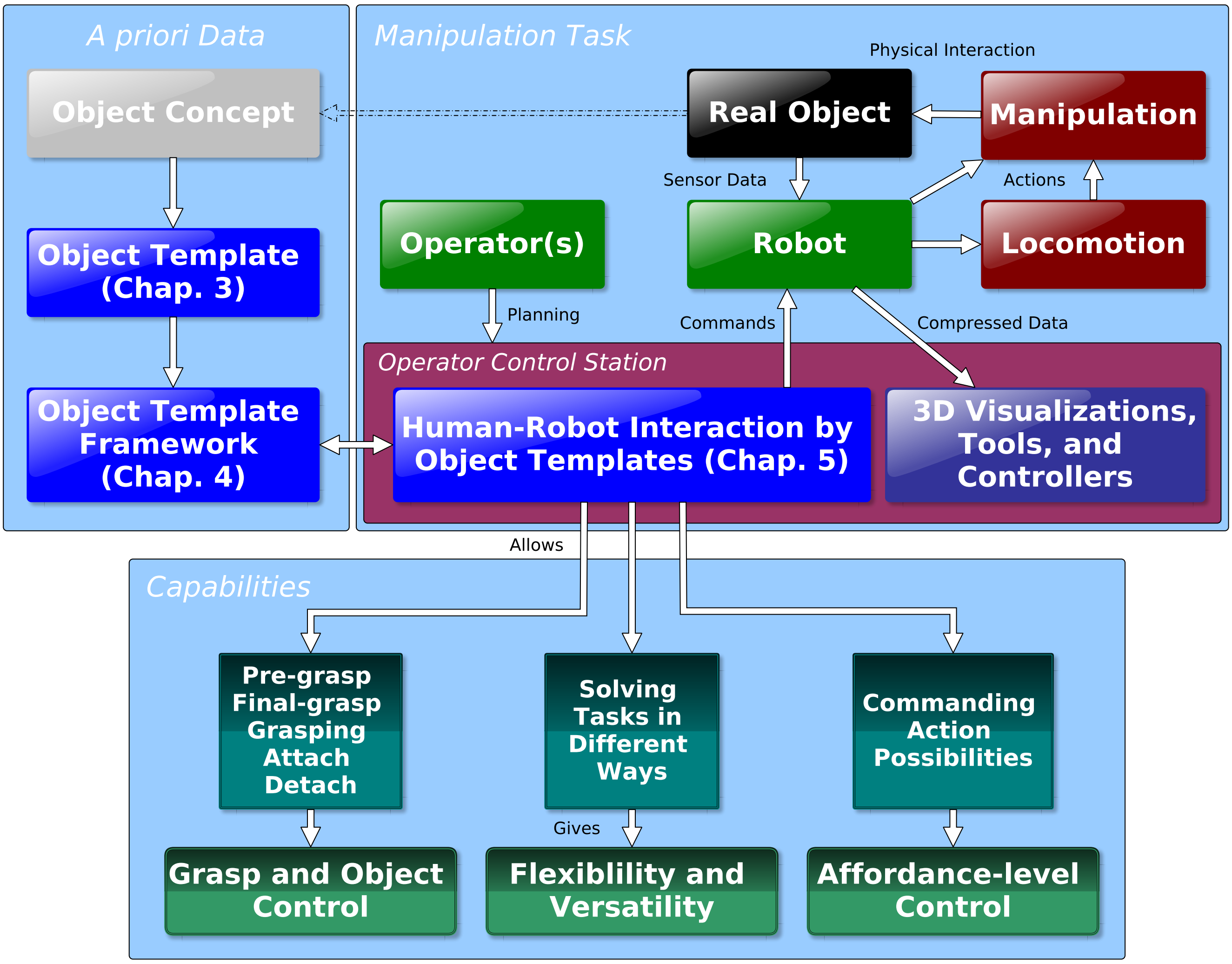


Figure 1.2: This graphic describes in a general form the components of the object template approach to manipulation for semi-autonomous avatar robots.

1.1 Context

The events that occurred in 2011 during the nuclear disaster at the Fukushima Dai-ichi Plant in Japan, showed the lack of robotic technology to help in these situations. As an initiative to motivate the development of remote robots capable of helping humans in cases of disaster, the Defense Advanced Research Projects Agency (DARPA) proposed in 2012 the renowned DARPA Robotics Challenge (DRC) competition. The DRC objectives pushed the development of robots and made them capable of remotely performing rescue tasks by simulating in real word scenarios sub-tasks such as manipulating fire hoses, turning valves, using tools and removing debris, as well as mobility tasks.

The research topics presented in this thesis are related to the DRC objectives and thus it was used as testbed for development and experimentation. The author participated in this competition as part of the international Team ViGIR [109]. Additionally, this object template approach was also provided to two other teams, Team Hector [107] and Team Valor [113], which also participated in the DRC.

1.2 Contribution and Contents

The goal of this thesis is to provide a method of interaction that can improve the execution and success of a remote manipulation task by allowing an efficient communication of a human operator's intent to a semi-autonomous avatar robot.

This thesis contributes in different aspects with capabilities that are required for human-robot interaction to perform remote manipulation tasks and investigates results from experimental evaluations while using an approach based on object templates.

- An object template concept as a representation and abstraction of the information of a real object that can be used and understood by both humans and robots. This object template concept adds to the current state-of-the-art approaches the possibility of describing the use of objects at the affordance-level as well as providing physical information required for manipulation.
- A framework for organizing and using the information abstracted in object templates based on three general independent-blocks of information: Object Template Library, Grasp Template Library, and Stand Template library.
- A manipulation interface to allow interaction between a human operator and a remote robot using a virtual environment for commanding actions at an affordance-level additionally providing different levels of object control.
- Experimental evaluation and validation of the approach performed in highly challenging scenarios. It has also been demonstrated that the approach is applicable to different robot architectures by its use on different humanoid robotic systems.

These contributions are elaborated in this thesis in different chapters as follows:

Chapter 2: Background of Remote Manipulation Robots and Object Manipulation Theory

In Chapter 2 a brief overview of the state of research in manipulation robots and object manipulation theory is given. Common aspects of remote manipulation robots are described in three different groups regarding their specific properties. First, pure teleoperation is described and examples of teleoperated robots are given. Second, state-of-the-art examples of robots operating in a full autonomous way are presented. Third, recent approaches considering an operator in the loop while performing remote manipulation tasks are described. Finally, object manipulation theory is described using the concept of "affordances" from the perspective of the robotics community.

Chapter 3: The Concept of Object Templates

In Chapter 3 the concept of object templates is introduced and a detailed definition is given. Here it is described how abstracting and compiling relevant aspects of objects into an entity of information can be used by humans and robots. Providing this information to a remote robot contributes to a better performance in a manipulation task since autonomous capabilities can be leveraged to execute different sub-tasks such as motion and locomotion planning. This chapter also gives a brief overview of related state-of-the-art concepts and describes how the object template concept proposed in this thesis goes beyond the state of the art.

Chapter 4: Object Template Framework for Remote Manipulation Control

Based on the proposed object template concept, a framework to organize, define, and provide object information to operators and the robot is developed in Chapter 4. This chapter describes the three main blocks of information for object templates: the Object Template Library which contains robot-agnostic information of the object of interest, the Grasp Template Library which includes information that describes poses and postures of a particular end-effector used by the robot, and the Stand Pose Library which describes the poses where a humanoid robot can stand to be able to reach that object using that particular end-effector.

This chapter also describes a particular aspect of the object template concept proposed which allows a human operator to improvise in a manipulation task. In this approach, improvisation refers to the capability of transfer manipulation skills defined for one object in a particular object template and using this object template to manipulate a different object. This interesting ability to transfer manipulation skills from known objects to new or unknown objects allows adapting to unforeseen situations. Being able to perform a manipulation task in versatile ways increases the potential of achieving the tasks objectives.

Chapter 5: Human-Robot Interaction Through Object Templates

Chapter 5 describes the design and development of an interface that allows a human supervisor to interact with a remote robot using the proposed object template concept. Specific aspects of the complexity of commanding a remote robot to manipulate an object are described. These aspects consider locomotion to approach to the object of interest, previewing and executing robot arm poses to move the end-effector to the desired grasping pose, finger control of the end-effector, allowing the object template to be attached to the kinematic chain for collision avoidance, selecting pre-defined points of interest to be used in the object, and executing manipulation motions at an affordance-level. Additionally, the proposed object template concept contributes to a collaborative autonomy approach. The previously mentioned aspects can be executed by a human supervisor or systematically executed by a high-level behavior.

Chapter 6: Experiments, Applications, and Results

Chapter 6 describes the experimental evaluation conducted to validate the contributions made by this thesis. The particular aspects of the object template concept, the framework provided, and the interface to provide this information to the remote robot are tested in different scenarios. The capabilities of the approach are demonstrated during participation in renowned robotics competitions and also through laboratory investigations. Additionally, this chapter describes experiments performed by another research group in a different country and with a different robot hardware using the concept and software developed in this thesis.

Chapter 7: Conclusion

This thesis concludes by summarizing the contributions to the state of the art. Here also an outlook of future directions for research and development is provided.

2 Background

In this chapter, a brief introduction to existing approaches used for controlling remote manipulation robots is presented. Normally, these approaches focus on either pure teleoperated or fully autonomous robots. Instead of focusing on one of these extremes, in the last years, new autonomy-centered approaches have emerged that investigate the middle ground as a way of dynamically variating the degree of autonomy that a system has. These approaches combine both, pure teleoperated and fully autonomous approaches, to leverage their strengths and weaknesses. Some of these approaches are sliding autonomy [10], adjustable autonomy [48], and human supervision [70], [118]. Other recent approaches such as Coactive Design [38, 9], propose a teamwork-centered design that focuses on managing the interdependence of joint activities between agents in a system. Once the interdependence relationships in a system are understood, the implementation of agent capabilities can be shaped to provide a better interaction.

Even though these approaches differ from each other, they are based on the same control loop primitives: SPA.

- **Sense:** A robot obtains extrinsic and intrinsic information using its available sensors. Information from the environment is commonly acquired by the use of cameras, laser range finders, and force/contact sensors among others. Information from the robot state is acquired by the use of an Inertial Measurement Unit (IMU) and joint encoders among others.
- **Plan:** Information obtained from the sensors needs to be processed in order to decide and make choices of future actions. With this information, for example, a clear path for locomotion can be found or an object required for manipulation can be recognized.
- **Act:** Once a plan has been formulated, the robot can perform actions to interact with the environment and make the necessary changes to accomplish a goal.

Sensing and acting are processes that in the context of this thesis are strictly required to be performed by a remote semi-autonomous robot. However, planning can be subdivided in a hierarchical decomposition which subtasks can be performed either by a human operator or by autonomous processes in the robot. This is called robot language hierarchy and consists of the following levels: System–Task–Action–Robot–Joint–Physical [57]. While pure teleoperated approaches focus in giving a human control from system-level to joint-level and letting the robot the only task of physically interacting with the environment, full autonomous approaches focus in giving the robot control from task-level to physical-level. This thesis presents an approach that aims at enabling a robot to operate at an action-level (and potentially at task-level if an autonomous behavior is used) while letting a human operator to dynamically change the interaction level to support the remote robot. An early attempt to automatically generate action sequences from task specifications was developed by Standford, it was called the Stanford Research Institute Problem Solver (STRIPS) [26].

Sections 2.1.1, 2.1.2, and 2.1.3 discuss how the distribution of these primitives between a human and a robot during a remote manipulation task contributes to different characteristics of the approaches described in this chapter.

This chapter also introduces background of the theory about how objects in the environment present properties to subjects for interaction and how this theory of interaction has been applied to the robotics community in Section 2.2. The presented thesis concept is based on this theory and proposes a distribution of the SPA paradigm to provide an interaction method for remote manipulation robots. The chapter concludes with a discussion about how manipulation control of remote semi-autonomous robots by the presented concept addresses the current needs of the existing approaches.

2.1 Remote Mobile Manipulation Robots

2.1.1 Background from Pure Teleoperated Manipulation Robots

Remote environments which have been degraded by a natural or human-made disaster might be dangerous for humans to explore. For this reason, teleoperation of remote robots has been used for Urban Search and Rescue (USAR) tasks in the last years. As shown by the work of Nagatani et al. [66, 65], the use of teleoperated robots or Unmanned Ground Vehicle (UGV)s can help to obtain information of hazardous environments (see Figure 2.1). In their project, they use a wired network to overcome communication issues, but this also leads to limited exploration range and risk of entanglement. During the mission of the robot Quince to the Fukushima Daiichi nuclear plant in 2011, the access to the third floor was blocked by rubble; semi-autonomous manipulation of such unstructured objects could have led to continued exploration.

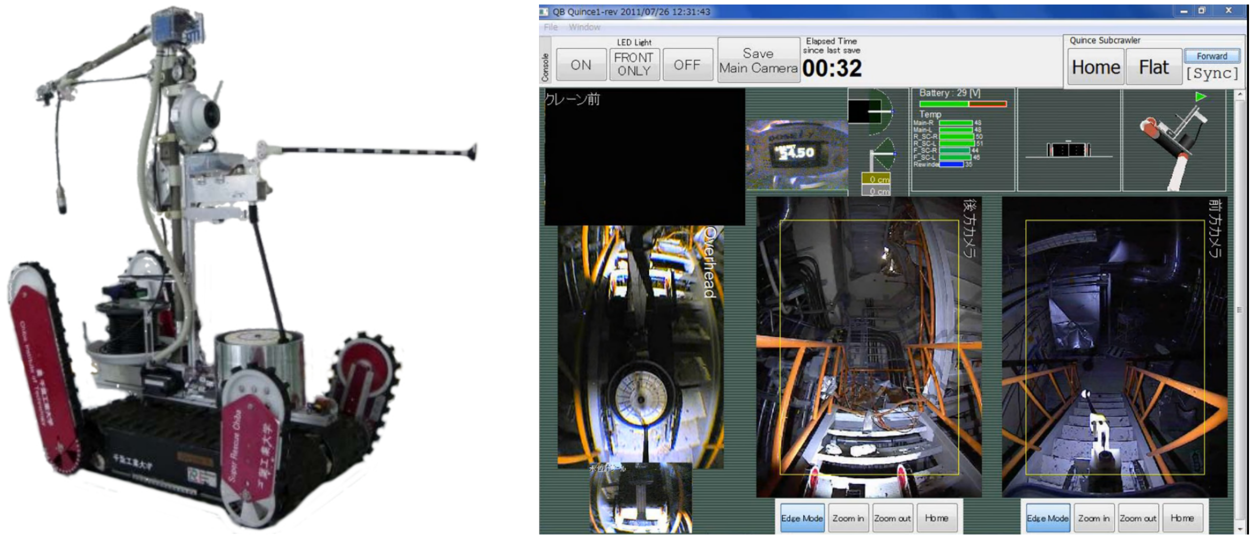
Teleoperation of UGV systems is mainly done using image feedback via cameras onboard the robot. Using only images, operators are put under tremendous mental workload which provokes stress [97] and can lead to failure of robot operations.

Pure teleoperation of humanoids is a complex task given the high number of degrees of freedom that need to be controlled [32]. Since per-joint teleoperation is not desirable, alternative master-slave approaches have emerged to teleoperate humanoid robots. For example, the European Space Agency (ESA) has developed an exoskeleton that allows a human to teleoperate a robotic arm [93, 75]. In this approach they also consider an intuitive user interface that allows for bilateral teleoperation [73]. The last up-to-date most interesting teleoperation experiment was performed by the astronaut Andreas Mogensen who commanded a robot to execute a manipulation task in Earth while being in the International Space Station (ISS) wearing this exoskeleton [1].

Force feedback and impedance control are also desirable characteristics while teleoperating a robot, for example, the tele-manipulation with the joint impedance regulation approach presented in [11].

2.1.2 Background from Full Autonomous Manipulation Robots

Fully autonomous robots have the advantage of not needing communications and that they are independent from a human operator. While a lot of research has been performed for autonomous humanoid robots in structured environments with impressive results [67, 92, 2], fewer results have been obtained for autonomous humanoid robots in unstructured environments.



(a) The Quince robot [66]. **(b)** Teleoperation Interface [29].

Figure 2.1: Teleoperation robot and images from the user interface used while exploring the Fukushima Dai-ichi nuclear plant.

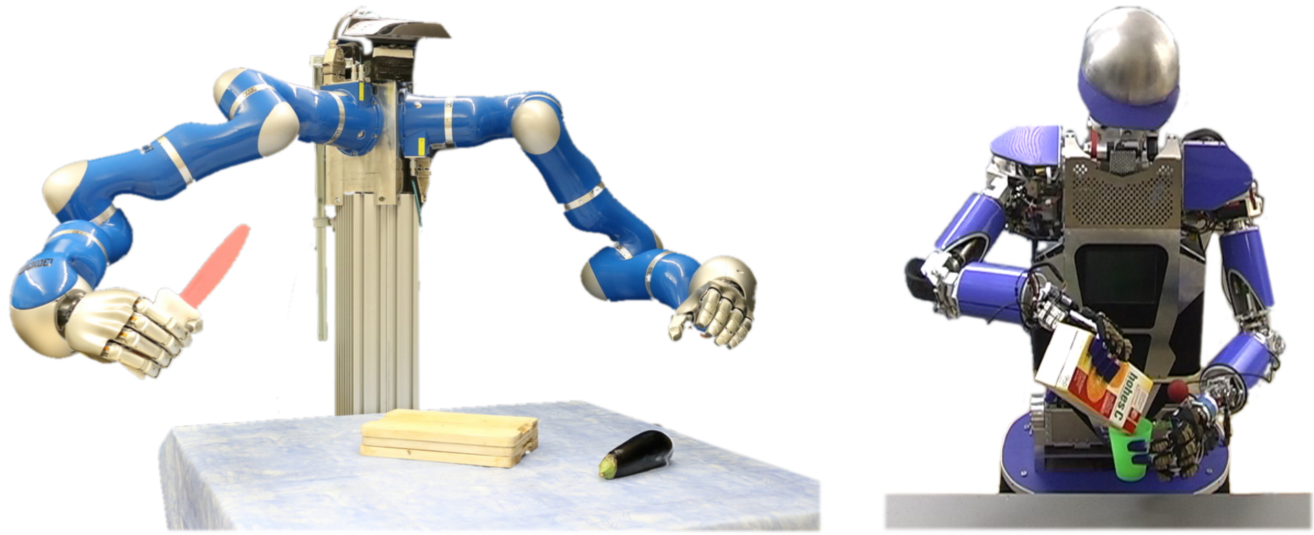
Kitchen environments are a very common application where highly autonomous humanoid robots are being developed for service [8, 5] (see Figure 2.2). These environments are expected to have different objects, however, the general structure of having stable lighting conditions and objects located above support surfaces is generally expected. Object recognition in these structured environments has led to development of autonomous approaches, for example, to allow a humanoid robot to acquire visual representations of the objects of interest [117]. Some approaches have contributed with comprehensive robotic systems that consider aspects in hardware as humanoid robots and software with manipulative, perceptive and, communicative autonomous skills [2, 3].

Other approaches use shape primitives to autonomously recognize objects and generate hypothesis of where the objects can be grasped [67]. Autonomous grasping approaches consider local symmetry properties of objects of interest in order to generate candidate grasps which are then tested for force-closure properties [72].

An approach from the Deutsches Zentrum für Luft- und Raumfahrt (DLR) for providing a fully autonomous robot with symbolic and geometric information of a task has been presented in [53].

Other approaches focus on learning methods for manipulation. For example the approaches presented in [54, 49] generate sequences of motion primitives and a hierarchical multi-phase to learn skills required to achieve a manipulation task.

The uncertainty of unknown environments, either because they are unexplored or because the have been degraded during a disaster, provides very little knowledge that current autonomous approaches can use. This presents challenges that prevent autonomous applications from using prior knowledge of the objects required to perform a task. Full autonomy is subject to challenges such as object recognition and mission planning, which given the conditions that can be found in unstructured environments, make fully autonomous robots not yet feasible to perform these tasks efficiently within the next few years.



(a) The Darius Robot [54]. **(b)** The ARMAR-III Robot [3].

Figure 2.2: Full autonomous robots.

2.1.3 Background from Semi-autonomous Manipulation Robots

Semi-autonomous robots have emerged as a middle-ground alternative between pure teleoperated and full autonomous robotic approaches. This approach focuses on leveraging the strengths from pure teleoperated and full autonomous approaches while minimizing their weaknesses.

From one side, humans have the ability to abstract information from the environment and understand the situation and requirements to perform a task in an unknown environment. This is supported mainly due to the high perception abilities that humans have to recognize objects in the environment. Once requirements and elements in the environment have been identified, strategic solutions can be formulated.

From the other side, robots have high computation abilities that can solve complex problems quickly. For example, obtaining inverse kinematic solutions for a high number of DOF, calculating a path for locomotion free of collision, and generating constrained manipulation motions among others, are commonly well known solved problems that for a remote human operator are highly challenging.

With a semi-autonomous approach, where a human will interact with a remote robot, new challenges arise. A particular interest of an interaction method with remote semi-autonomous robots is the ability to transfer the human operator intent to the remote robot, converting it into actions in order to perform tasks in the environment. In order for a robot to perform manipulation tasks, it needs to acquire information about the objects to be manipulated, such as physical and abstract information. A human operator performing perception tasks, for example, can provide to the robot with more reliable information about the environment than it can be obtained autonomously.

Considering a human operator in the loop for remote manipulation tasks has been proven as an efficient approach to deal with challenges that uncontrolled and potentially degraded environments present. A discussion of the semi-autonomous approaches of renowned research groups such as the Massachusetts Institute of Technology (MIT), National Aeronautics and Space

Administration (NASA), DLR, and the Institute for Human and Machine Cognition (IHMC) will be detailed in Section 3.1.

Even though none of the semi-autonomous approaches discussed in this thesis have been used in real disaster operations, they have shown impressive results in real world simulated disaster scenarios. The research group at Korea Advanced Institute of Science and Technology (KAIST) participated in the DRC with the humanoid robot "DRC-Hubo" shown in Figure 2.3a. Team KAIST ranked first place in the DRC demonstrating locomotion and manipulation abilities performed in a semi-autonomous way. In the same way, the team Tartan Rescue from Carnegie Mellon University (CMU) which ranked third place showed impressive robot capabilities with their robot "Chimp" shown in Figure 2.3b. As it can be seen in Figure 2.3, these two teams' hardware was able to perform bipedal locomotion tasks, however, a wheeled transformation alternative was used during the DRC allowing for easier locomotion over flat terrain compared to pure humanoid robots.

(a) The DRC-Hubo Robot. **(b)** The Chimp Robot.

Figure 2.3: Semi-Autonomos robots. Images taken from DARPA[18].

2.2 Object Manipulation Theory

Object manipulation has been a strongly researched problem in the robotics community. To try to overcome the challenges of identifying the potential use and manipulation skills required to use objects, some researchers have studied the concept of affordances.

2.2.1 The Concept of Affordances

The term "affordances" was first introduced by the psychologist J.J. Gibson in [30]. An affordance, from the perspective of Psychology, is a term that describes the possible actions that an object offers to an organism in the environment. An affordance is not a physical property and it depends on the organism perceiving this affordance.

For example, a flat, extended, and rigid surface can offer support-ability for a cup of coffee if it is horizontal; if it is vertical it can offer supportability to a person leaning on it. However, if that vertical, flat, extended, and rigid surface can also rotate on a single vertical axis at one edge “a door”, offers an additional affordance, for example, close-ability of a room.

The concept of affordance has been adopted by research fields related to robotics as an approach to define how a robot should behave in order to accomplish a determined task using an object. In this context, affordances describe the relationship between an object and the possible actions that a robot can execute to manipulate this object.

A powerful characteristic of affordances is that from the perspective of achieving a defined task, the object used is irrelevant, as long as it provides the affordance required to fulfil the task. J.J. Gibson wrote: *”If you know what can be done with a graspable detached object, what it can be used for, you can call it whatever you please”*. This characteristic provides the ability to improvise, or in other words, the potential of accomplishing a specific task using objects that differ to the ones designed originally for that task.

2.2.2 Affordance-based Applications in Robotics

Şahin et al. [90] presented an extensive work to formalize the term of affordance in the robotics field. In this work it is proposed that affordances should be analysed from three different perspectives: *observer*, *agent*, and *environment*. This formalization has been key to the development of robot control approaches, e.g., [6, 117]. The following paragraphs also describe some concepts and applications in which affordances have influenced development in robotics.

Object Action Complexes

Kruger et al. [50] proposed the concept of Object-Action Complexes (OAC), which aims at defining the relationships between objects and actions. OAC is an action-centered concept created to formalize this relationship and it has been used to develop control approaches that aim at allowing robots with the possibility to understand how an object will behave after an action is performed over it. OAC allow autonomous robots to learn and predict the behaviours derived from performing an action over an object as well as to build symbolic representations of continuous sensorimotor experience.

Locomotion and Manipulation

Autonomous extraction of affordances has been investigated in [40] and later in [41] for whole-body motions. In this work, a humanoid robot is considered and the challenges that bipedal robots face are taken into account when the extraction of the affordances in the environment is done. Using reachability and stability maps, the space of affordances found is limited to the ones that are directly usable by the robot. In this work, affordances are constrained to consider a predefined set of rules that links symbolic affordances to physical properties of the extracted primitives like plane orientation, principal axis, or extent of a surface.

Grasping

Grasp planning for unknown objects is an application where the concept of affordances has been widely used. ten Pas et al. [111] proposed finding grasp poses not by recognizing and searching objects directly in a database but by recognizing *“a geometric characteristic of an object that allows it to be grasped by a particular robot hand or gripper”*.

In [6] grasp affordances for previously unknown objects are found through tactile exploration. In this work, pairs of object features such as planar surfaces (not considering edges and vertices) are used to investigate potential grasp poses for a planar gripper.

Task Communication

Communicating task actions to robots has also been explored from the perspective of affordances [34]. In this work, affordances are defined as action possibilities and task communication is centered in an object-action context. This indirect task communication works by communicating only one of two parameters, either the action or the object. Thus, it is assumed that a full autonomous robot possesses knowledge of which actions can be performed with each object. In this work an interesting concept regarding ambiguity is presented. In an ambiguous environment, objects can be associated with different actions, and actions can be executed using different objects. This ambiguity concept for full autonomous robots is related to the improvisation ability that a human operator has to command the robot to perform actions with different objects and vice-versa that this thesis describes as presented in Section 4.7.

In a work presented by Moratz et al. [59] affordances are described as functional object aspects that are visually perceivable. This affordances are analysed considering the spatial relationships that actions require. They are created by the designer of the recognition system and are shared between the robot and operator.

Tool Use

In the last years, several research groups have made important contributions to the use of tools in robotic manipulation. The concept of *tool affordances* has been introduced in different research approaches [101, 99, 102]. These approaches use behavior-grounded concepts to allow a robot to learn a compact predictive model of the tool affordances. A definition of tool affordances is given in [37] as the ability of the robot to be aware of the possible actions and effects that a tool can create in the environment. Another interesting approach is the utilization of objects as tools. Stilman et al. [100] proposed a "MacGuyver" paradigm for manipulation robots saying that robots should also take advantage of the objects present in the environment to achieve a task.

More recently, tool affordances have been explored from the perspective of computer vision to autonomously recognize and identify uses and points of interest in objects [114, 31, 64]. However, since these approaches are designed for fully autonomous robots, the tools used required specific colors that contrast with a well defined background color, situation that is not common in uncontrolled environments. An approach that identifies points of interest on objects and transfers manipulation skills to similar objects using the same points of interest as matching features is presented in [103]. This approach presents an interesting method to adapt tool use to new similar objects, however, there is little detail on how the points of interest in objects are used for planning manipulation skills.

2.2.3 Discussion

The term of affordance has been a complex term to define. As initially defined by J.J. Gibson, an affordance is an economical perception of the relevant information of the environment without the need of perceiving the whole world [30]. As described in Section 2.2.2, in the robotics community there have been several perspectives into which affordances have been defined and applied. For the purposes of this approach, an affordance is considered as the potential use of an object in the environment that can be actuated using the available manipulation skills of a robot. As an example, if we consider the affordances of a door, it includes *turnability* of the handle, *openability*, and *walk-throughability*. Affordances like *leanability* (for supporting) or *hideability* (e.g., behind it) are not under the scope of this approach.

3 The Concept of Object Templates

This chapter presents first a brief overview of a subset of related work presented in Section 2.1.3 with focus on concepts similar to the concept of object template. Afterwards, a definition of the object template concept in the context of this thesis is presented.

3.1 Related Work

In this section a review of other concepts for manipulation control of remote robots is presented. Like the approaches presented in Section 2.1.3 most of current semi-autonomous robotic approaches use 3D geometry meshes from the objects for providing grasping information, stance information and require the operator to move this meshes to generate trajectories for joint motion. Other approaches present templates that define manipulation instructions that can be interpreted in a symbolic level and be grounded to a geometric level for a task. The following approaches are representative of the state of the art in remote manipulation of robots.

Affordance Template ROS Package

The Affordance Template (AT) ROS Package [33] presents a shared autonomy approach for remote human-robot interaction. Devolped by TRACLabs and NASA, this package defines human-adjustable robot task waypoints with respect to an object frame of reference. These predefined end-effector waypoints are used as targets for grasping and manipulation of the objects they represent. These waypoints are used to generate and execute trajectories once the template has been manually registered to the sensor data in the user interface. Using this package, a human operator can adjust template scales to adapt to new similar objects with different sizes simultaneously scaling the predefined waypoints for manipulation.

This approach does not focus on considering physical information of the object such as mass or center of mass. The use of predefined waypoints prevents the operator from adapting to change the grasp pose of the end-effector on the fly. The concept of affordances is limited to this predefined waypoints and does not describe the general manipulation skill required to use the object. This limits flexibility and prevents to solve manipulation tasks in versatile ways.

Action Templates

Human supervision of remote manipulation tasks is not limited to rescue applications. For example, an approach to provide an astronaut with an interaction method to command a robot for space application has been presented by the German Aerospace Agency in [7]. They present the concept of Action Templates [53] to define descriptions of specific robot actions. This approach categorizes objects in classes according to their functionality in a hierarchical structure. These functional object classes augment the object definition with templates of actions that the robot can perform. Similar to the *object oriented paradigm* in computer languages, this approach enables inheritance of actions from general object classes to specific object classes. The *planning domain definition language* [56] is used to combine the symbolic level of a task (predicates and actions that describe the object states and their transitions) with the geometric level of a task (a description of the interaction with objects). This approach was initially developed for fully autonomous robots working on controlled conditions. Afterwards, the approach was extended as a tablet computer application for allowing shared autonomy between the remote robot and the human operator (Section 5.1).

While this approach provides an interesting concept on how to ground symbolic information of a task into robot actions, it provides a basic interaction method for the operator to manipulate virtual objects. While it is expected that robots perform accurate motions, imperfections in the execution of manipulation tasks in uncontrolled environments still requires human supervision. For these cases, this approach does not present an interface to provide the human operator with the ability to aid the robot to deal with inaccuracies. This approach does not focus in considering physical information of the object as part of their virtual objects. Also, the use of objects is limited to the manipulation information described with respect to the object frame of reference. An interface to define manipulation motions with respect to a point of interest in the object grasped is not considered.

Coactive Design: Interactable Objects

The concept of Coactive Design [38] focuses on designing a system considering interdependency between participants in joint activity. Derived from this concept, an approach for an operator to control the behavior of a humanoid robot called *interactable objects* was presented in [47] and used by the Institute for Human and Machine Cognition (IHMC) which placed second in the DRC Finals [39]. These interactable objects allow an operator to transfer action intent to the robot, for example, by selecting different grasp poses for the end-effectors, selecting different manipulation stance poses that allow the robot to reach the object, and also gives the ability to transfer information about how to perform footstep plans for locomotion with respect to the object of interest. To transfer information about how to manipulate objects in the environment, end-effectors can be linked to the interactable objects. This way, when the operator modifies the pose of the interactable object, the end-effector follows that pose and a trajectory is generated which can then be sent to the robot to execute it.

Online Affordance-based Perception

In an approach presented by MIT [23], Computer Aided Design (CAD) models of objects are used as environmental features that allow an operator to command robot actions based on the action possibilities that the real objects have. These templates provide information such as manipulation stances where the robot is able to reach the object and predefined potential grasp locations. In this approach, to generate arm trajectories, the operator is required to manipulate the object template and change its pose. The robot end effector follows the pose of the template and this information is used to define a desired end-effector trajectory. They present examples of performing plan motions with respect to a point of interest in the drill, such as the “drill bit”. To the best of the author’s knowledge, the approach presents no option to the human operator to select different points of interest from the objects grasped on the fly.

3.2 Contribution

The concept of object template presented in this thesis systematically extends other state-of-the-art approaches by considering additional information and capabilities. This concept of object template, additional to the shape, stance, and grasping information, provides information about the physical properties of the object (used for control and motion planning), as well as abstract information related to the affordances of the object and points of interest on the objects. This information improves the communication of the human operator intent into robot action. For example, the operator can select to use the affordance of the object and let the robot generate joint trajectory motions autonomously. Additionally, the definition of usabilities (Section 4.5) allows generation of manipulation motions with respect to specific points of interest on the real objects which enables these objects to be considered as an augmentation of the end-effector of the robot.

Particular aspects of the object template concept presented in this thesis include:

- High-level interaction between the human supervisor and the remote robot by commanding actions at an affordance-level which increases the efficiency of planing and motion execution.
- The ability to define points of interest on an object to be used for motion planning which allows objects to be considered as part of the end-effector increasing the reachable range of the robot.
- Flexible means of interaction by allowing the use of the affordances on similar objects or manipulation tasks of the same class as well as the use of intermediary objects.
- Allowing the human operators to use the improvisation ability by identifying versatile ways of achieving a manipulation task through the use of objects in different ways or with other purposes.

Multiple people have contributed to the object template concept. These contributors are named here: Stefan Kohlbrecher and David C. Conner (initial concept), Felipe Bacim (3D visualization), and the author contributed with the systematically extension of the concept to consider additional physical and abstract information of the objects as will be described in this chapter.

3.3 Definition of an Object Template

A condensed version of this Section was published in: 2014 14th IEEE-RAS International Conference on Humanoid Robots (Humanoids) [76].

An object template is the abstract representation of a real object that allows a human operator to interact with a remote robot to perform manipulation tasks. This means of interaction is based on the idea that the higher the level of abstraction in the communication between humans and robots the more efficient a manipulation task can be performed. For this reason, the need of an entity of information that can be understand and easily manipulated by humans and that at the same time a remote robot can use to perform manipulation tasks arises. The purpose of an object template is to allow a human to describe the functionality of an object including its kinetic and kinematic properties, and that this information can be interpreted and transformed by the remote robot into joint motions for manipulation.

Object templates are designed to increase the level of abstraction in the communication between a human operator and a remote robot because they contain information of the functionality of how the object needs to be manipulated which is related to purpose of the object. The functionalities that an object in the environment offers to a subject are known as *affordances*.

Affordances were introduced by the psychologist J.J. Gibson in 1977 [30] (Section 2.2.1), but it has been adopted by the robotics research community to help in the definition and understanding of object purposes and manipulation skills that robots require (Section 2.2.2). Section 3.1 identifies and describes different examples of state-of-the-art cases in the context of remote manipulation robots and their interfaces.

Object templates of known objects are created to be able to provide fast and efficient information of the objects of interest to the remote robot. They are designed to be a general shape of the real object so they can be used for similar objects (e.g., drills from different brands). These object templates contain additional information about each object such as physical and abstract information. Object templates are created to provide pre-computed potential information to aid a robot with manipulating the real objects. This information considers robot poses with respect to the object to facilitate reaching and grasping, pre-grasp and final-grasp poses for the end-effectors, finger postures in the end-effector for grasping, information about possibilities of action with the object, as well as information about points of interest on the object.

Using object templates, the human operator can aid the robot to identify objects of interest in cluttered sensor data as well as their respective properties. Object templates contain valuable information of the object they represent which is acquired by a manual offline analysis the objects. A database of this information from several objects is created and used afterwards during online execution of the system. Physical information of the objects could in principle be autonomously learned by estimation of the properties, for example, using perception algorithms for visual properties and analysis of the information acquired by force-torque sensors for inertial properties. However, acquiring this information autonomously can be error-prone, for this reason in this approach the information is manually acquired offline.

3.4 Information Contained in an Object Template

3.4.1 Physical Information

Object templates provide information about the physical properties of the object since some of them play an important role while manipulation is being performed. Properties such as shape, mass, Center of Mass (COM) are highly relevant, while properties such as odor, taste, and electrical conductivity are not relevant for manipulation. This information should describe the internal properties, the external properties of the object, as well as the location properties of the object with respect to the robot.

Internal Information

Internal physical information of an object refers to properties of the object such as mass, COM, and inertia tensor. Providing this information to the robot is useful when the actuator controller of the robot can consider external forces. In principle, additional physical information can be included in an object template, e.g., density and hardness. However, for the purposes of this approach only the mass, COM, and inertia tensor, are considered since they can be used to calculate dynamics when the kinematic chain of the end-effector is augmented with the grasped object (see Table 3.1). Friction information could also be included for grasping purposes, however, it is not currently considered in this approach.

External Information

Object templates should reflect the general shape of the real object that they represent. For example, using a virtual representation, real objects can be visualized as a 3D geometry mesh in a computer environment. Shape information is important when collision avoidance is required during manipulation tasks. Additionally, object templates should be able to be considered as part of the robots end-effector after grasping, and using the shape information, motion planners can then consider the object shape to generate trajectories that prevent collisions between the real object and the environment.

Object	Drill
Mass (Kg)	2.4
COM (m)	(0,0,0)
Bounding Box Min (m)	(-0.10,-0.04,-0.12)
Bounding Box Max (m)	(0.14, 0.04, 0.13)

Table 3.1: Physical Information of the object.

Figure 3.1: Drill: DeWalt DCD980M2.

Location Information

This information is the most used when supervising a remote robot in an unstructured environment. This information is not explicit in the definition of an object template, it is implicit when the object template is used. Analysing the sensor data provided by the remote robot, a human operator can have an insight of where and how the objects of interest are located in the environment. For humans, this estimation is often easy, but for robots, acquiring this information autonomously can be error-prone. For this reason, a human operator can identify the pose (position and orientation) of an object and use this information to place the 3D geometry mesh of the object template in the virtual environment. This will provide the robot with an estimated information of where and how the objects required for manipulation are located in the environment (see Figure 3.2).

3.4.2 Abstract Information

Object templates should also provide information about how to use the object. This information will help the robot to perform locomotion and manipulation in situations where autonomously perceiving objects of interest gets challenged by environmental characteristics such as cluttered scenarios, lighting, and degraded objects among others.

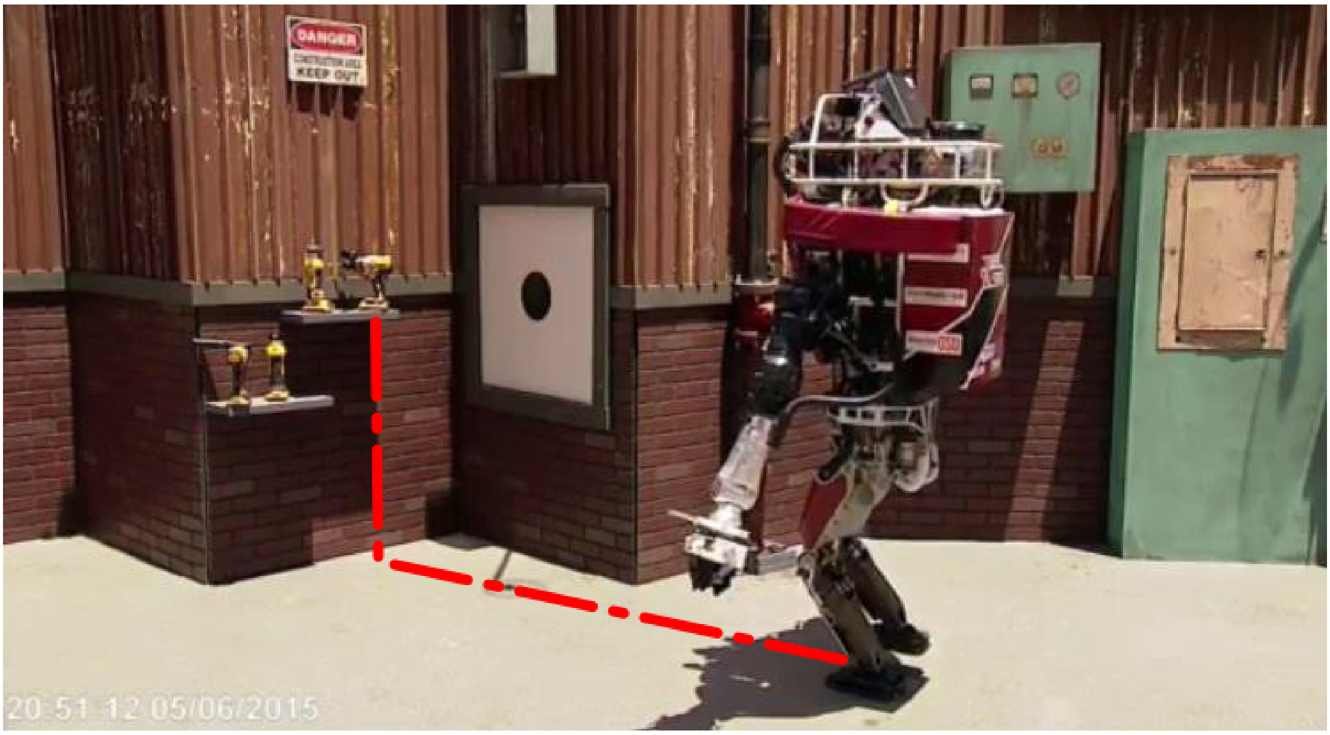


Figure 3.2: External physical information of an object. The pose (position and orientation) of an object located over a table can be estimated with support of the human operator using the sensor data.

Stance

Object templates include information about potential standing poses which will allow the robot to reach the objects. Using this information a human operator can command the robot to perform locomotion and move towards a predefined stand pose described in the frame of reference of the object template. Solving this problem autonomously has been previously researched as inverse reachability [115],[13]. However, they do not focus on sensor visibility constraints or control-related constraints due to appendage control performing better in some configurations than others. For this reason, in this approach, an offline empirical analysis of potential stand poses can be made and this information is preloaded and used during online operations. In [104] an example of performing autonomous locomotion and footstep planning with collision avoidance using information provided by object templates is presented.

Grasp

Object templates include information about potential grasp poses for the end-effectors. Using this information a human operator can command the robot to perform arm motions to reach the object at a predefined pose which will allow the robot to grasp the object. This grasp information can be defined in two different poses: a pre-grasp pose and a final-grasp pose. The pre-grasp pose is used to create a target pose of the end-effector that lies on a vector at a short distance before the object. This helps to generate arm motions using collision avoidance algorithms which might have difficulties finding a trajectory towards a final position around the object.

The final-grasp pose is a target end-effector pose that allows the robot to take control of the object. To reach the final-pose, the robot needs to generate arm motions which will constraint the end-effector to move in a straight line between the pre-grasp pose and the final-pose.

Affordances

Object templates include information about potential use of the objects. This information is based in the concept of *affordances* which describe the possibilities of action that an object in the environment offers to a subject. This is the most important piece of information that the object templates contain, since information about how objects need to be manipulated is of high relevance to accomplish a task. Acquiring this information autonomously in structured environments has been previously researched in the robotics community (Section 2.2.2). However, acquiring this information autonomously in unstructured and potentially degraded environments is still an unsolved problem. For this reason, compared to other state-of-the-art approaches where only grasping waypoints are provided, the approach presented in this thesis additionally considers this information and describes the manipulation skills that are required to perform a specific task using this object. This manipulation skills define the specifics of a task, e.g., lifting, pushing, or turning an object. In principle, if this manipulation skills are well defined, they can be transferred between similar objects and manipulation tasks of the same class (Section 4.7). Detailed information about affordance implementation in the object template concept will be further given in Section 4.4.

Object Usabilities

Object templates include information about points of interest on the objects which are referred to as object *usabilities*. This information is relevant for manipulation when a specific part of an object requires to be considered when planning manipulation motions with respect to another object. A set of defined usabilities is generated offline, for example in a drill, the position of the trigger and the bit are points of interest that need to be considered for manipulation. During online execution of the system, a human operator can select between the different usabilities on an object grasped to generate motions with respect to another object. Detailed information about object usabilities and the implementation presented in this thesis will be further given in Section 4.5.

4 Object Template Framework for Remote Manipulation Control

In this chapter, an implementation form of the concepts described in Chapter 3 is presented. The objective of these concepts is to provide the remote robot with information about how an object in the environment should be manipulated. This implementation considers three different spaces into which the object information is divided: object space, end-effector space, and robot space.

The implementation presented in this thesis considers physical and abstract information of the real objects. However, compared to the manipulation approaches described in Chapter 2 this approach provides additional information about the manipulation skills required to perform a task. The two main concepts for representing manipulation skills are *affordances* and *usabilities*. These concepts have several advantages compared to the existing manipulation control approaches for remote supervised robots (Section 3.1). The objective of the proposed concept is to allow high-level communication with a remote robot for object manipulation. For this reason, this implementation of manipulation skills generalizes the motion constraints required to perform a task allowing them to be defined at a task level.

In order to define manipulation skills for an object, a classification of objects with respect to their mechanical constraints in the environment is presented. Also, a classification of manipulation skills that are most commonly used in manipulation tasks is given. This classification is based in the manipulation classification introduced in [12] and [25]. By using this manipulation classification, the manipulation skills designed for an object can be transferred to another object if they have similar properties, or if the manipulation skill belongs to the same class. This can be evaluated by letting the remote robot to execute manipulation skills based in a different object template than the real object that is being used.

4.1 Related Work

Several research groups have contributed to the state of the art related to robotic manipulation control. This approaches work as a planning backend to higher system layers that provide manipulation control at an action-level.

Drake

The MIT has developed the Matlab-based Drake dynamics toolbox [110]. This optimization-based planning, control, and analysis toolbox has been used as back-end for controlling nonlinear dynamical system, like the humanoid robot Atlas used during the DRC [23]. Drake provides a framework for whole-body planning motions for a high DOF robotic system such as a humanoid robots.

Affordance Template ROS Package

The AT ROS Package [33] presented in Section 3.1 uses the Affordance Template Description Format (ATDF) to store task specifications in an Extensible Markup Language (XML) format that is robot-agnostic. This package is implemented as a plugin for the tool interface “Rviz” which enables use with the Robot Operating System (ROS) framework. A database of affordance templates is stored and managed by an affordance template server which is located on board the remote robot.

Online Affordance-based Perception

The approach presented by MIT [23] in Section 3.1 defines the information contained in the templates in an XML file which they call Object Template Description Format (Object Template Description Format (OTDF)). In this file, objects are defined as a series of links and joints; the approach presented in this thesis defines the objects as meshes, for this reason this description format is not used.

Other approaches recently contributed during the DRC

In [16], a manipulation approach is presented focussing on some DRC tasks using the Atlas robot. In [4] a human-supervised manipulation control approach focusing on the door task at the DRC is presented.

4.2 Contribution

The contribution presented in this chapter is an approach for manipulation control based on the concept of object templates presented in Chapter 3. This approach allows a human operator to command an avatar robot to execute versatile object manipulation task at different levels of abstraction. From one side, using object templates, the operator is able to send information subject to low bandwidth constraints to an avatar robot. This information includes stand poses, grasp poses, and manipulation commands at an affordance level. It also considers the possibility to select and generate manipulation motions using predefined points of interest on the objects grasped that are particularly required to achieve a task. From the other side, this approach allows a human operator to improvise during a manipulation task by being able to transfer manipulation skills between similar objects or between tasks that belong to the same class.

This contribution allows achieving manipulation tasks utilizing objects in a different way than they were designed and with the possibility of executing manipulation with objects outside the reachability workspace of the robot, as will be demonstrated with laboratory experimentation in Chapter 6. Related publications [76], [77].

Multiple people have contributed to the object template manipulation control approach that the contribution within this thesis is part of. These contributors are named here: Stefan Kohlbrecher (manipulation planning backend), Felipe Bacim and Brian Wright Operator Control Station (OCS), and the author contributed with the library framework for object templates which will be described in this chapter.

4.3 Implementation of the Concept of Object Templates

For the purposes of this approach an Object Template Library (OTL) that can include any number of objects has been created. This accounts for potential unknown objects that might be available in a disaster scenario. The OTL is divided into three blocks of information: the object library (physical and abstract information of the object), the grasp pose library (end-effector pose information to grasp the object), and the stand pose library (robot stand pose information to move towards the object). The grasp pose library and the stand pose library have a relationship of many to one with the object library. Each object in the object template library has a unique type that is used to relate one or many grasps to one OT as well as for stand poses. An diagram of the relation between these libraries can be seen in Figure 4.1.

Figure 4.1: Relationship between objects, grasps and stand poses libraries.

The OTL has been implemented using the XML format because it provides a human-readable language for defining information. The following subsections provide a formal definition of each of this libraries as well as an implementation example in an XML file.

4.3.1 Object Template Implementation

An object template contains information about the physical properties of the object and abstract information about how this object can be manipulated. This information is robot-agnostic and grasp-agnostic. An object template is defined by the tuple:

$$o = (I, N, T, H, M, C, E, A, U),$$

where:

- $I \in \mathbb{N}$ is the ID number of the object of interest,
- $N =$ is the name of the object template,
- $T \in \mathbb{N}$ is the type of template (e.g., tools, debris, hose),
- H is the 3D geometry mesh of the object,
- $M \in \mathbb{R}$ is the estimated mass of the object,
- $C \in \mathbb{R}^3$ is the estimated COM of the object,
- $E \in \mathbb{R}^6$ is the estimated inertia tensor of the object,
- $A \in \mathbb{R}^3 \times \mathbb{SO}(3)$ is a frame of reference that defines the position and orientation (in quaternion form) of an affordance and it can be either a linear motion, a circular motion, or a combination of both motions if a screw pitch value $P \in \mathbb{R}$ is considered.
- $U \in \mathbb{R}^3$ is a point in the frame of reference of the object that defines a usability that is used to augment the robot's end-effector to consider this point of interest during motion planning.

The *Object Template Library XML* definition consists of the following tags:

1. <*templatelibrary*>: Is the root of the *XML* file.
2. <*template*>: Defines the *object template type*, the name and the group.

3. <*visual*>: Contains the mesh information.

4. <*inertial*>: Contains the internal physical properties of the object.

5. <*usability*>: Contains pose information from points of interest. Position defined by the *xyz* argument and rotation by *qx*, *qy*, *qz*, and *qw* arguments.

6. <*affordance*>: Contains pose information of the actions that can be performed with the object. Position defined by the *xyz* argument and rotation by *qx*, *qy*, *qz*, and *qw* arguments.

Listing 4.1: Short version of the object template for the DRC Drill.

```
<templatelibrary>
    <template name="DRC_drill" type="17" group="tools">
        <visual>
          <geometry>
            <mesh filename="path/to/drill.ply"/>
            <boundingbox min="-0.0760 -0.0407 -0.0030" max="0.1560 0.1407
                0.2543" />
          </geometry>
          <origin rpy="0 0 0" xyz="0 0 0"/>
          <material name="black">
            <color rgba="0.0 0.0 0.0 1"/>
          </material>
        </visual>
        <inertial>
          <mass value="1.465"/>
          <origin xyz="0 0 0"/>
          <inertia ixx="0.0001" ixy="0.0" ixz="0.0" iyy="0.0001" iyz="0.0"
              izz="0.0001"/>
        </inertial>
        <usability id="0" name="origin">
            <pose xyz="0 0 0" qx="0.0" qy="0.0" qz="0.0" qw="1.0"/>
        </usability>
        <affordance id="0" name="insert" type="cartesian" axis="z"
            displacement="0.05">
            <pose xyz="0 0 1.0" qx="0.0" qy="0.0" qz="0.0" qw="1.0"/>
        </affordance>
    </template>
</templatelibrary>
```

The mesh H is designed to be visualized in a virtual environment where 3D sensor data from the environment acquired by the remote robot is simultaneously displayed. This mesh is manually created by measuring the objects and are stored as a Polygon file format (PLY) or Stereolithography file format (STL) (see Figure 4.2). Currently, this approach considers static meshes, meaning that scaling an object templates to match the size of an object is not yet implemented.

The inertial properties of the object may be manually or algorithmically estimated if no reference data is available. The objects are weighed on a scale to find the mass and the COM are found by balancing the objects. Currently this approach does not use the inertia tensor of the objects. However, the functionality to consider the tensor of inertia is implemented.

External information of the object such as the pose of the object with respect to the robot can be estimated using an object template. In a virtual environment, an operator can place the

Figure 4.2: Cutting tool "Dewalt DCD980M2" (left), high detail mesh (middle), and low detail mesh of a similar drill including a handle (right).

object template over the sensor data that corresponds to the real object. This way, the robot can use the object template pose to identify the pose of the object (see Figure 4.3).

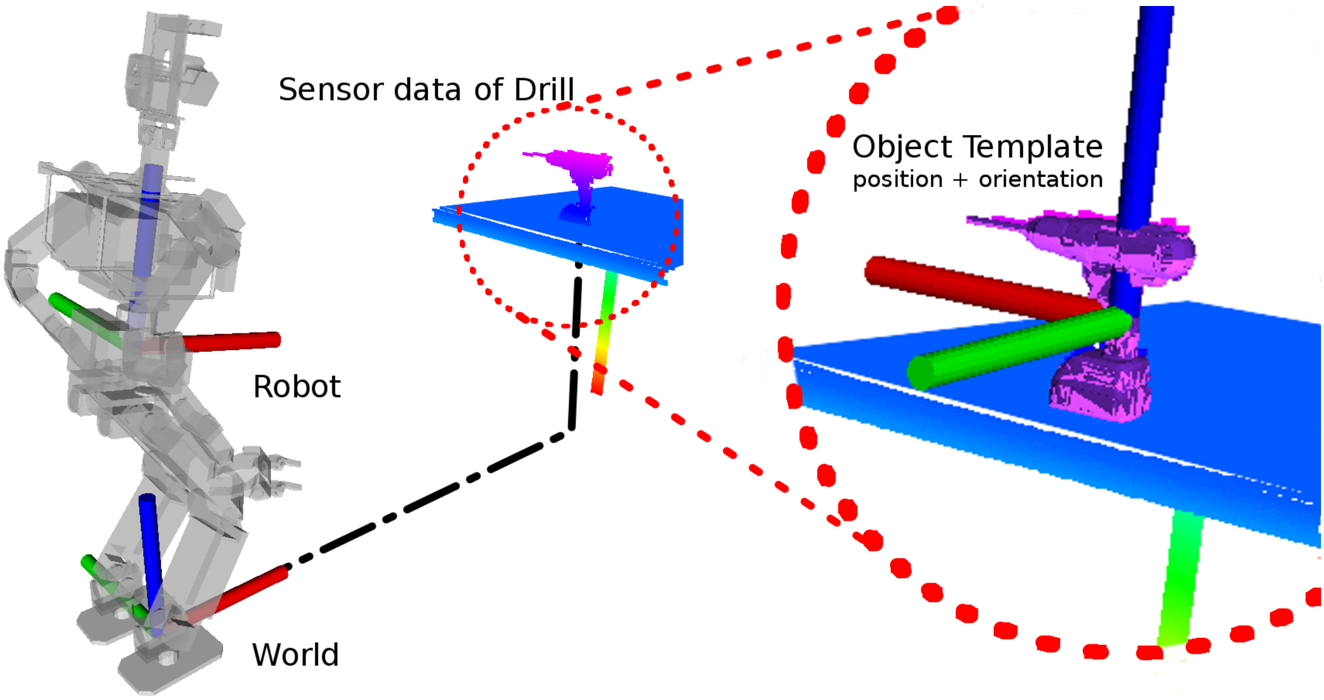


Figure 4.3: Example of an object template. The human operator identifies the sensor data of an object and places the object template that corresponds. The robot can then use the pose of the object template to estimate the location of a real object.

4.3.2 Grasp Template Implementation

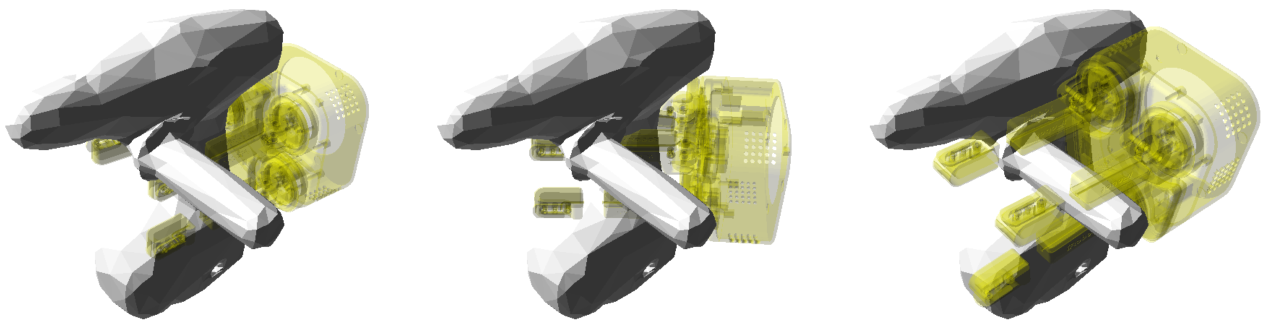
A grasp template contains information about a target pose and finger configuration that defines where and how an end-effector needs to be located to grasp an object template. This information is specific to the end-effector. Grasp templates are defined by the tuple:

$$g = (N, T, I, P_f, V_a, G_p),$$

where:

- $N =$ is the name of the object template that the grasp belongs,
- $T \in \mathbb{N}$ is the type of object template (e.g., tools, debris, hose),
- $I \in \mathbb{N}$ is the ID number of the grasp,
- $P_f \in \mathbb{R}^3 \times \mathbb{SO}(3)$ defines the pose (position and orientation) of the end-effector for grasping the object,
- $V_a \in \mathbb{R}^3$ is an approaching vector that the end-effector needs to follow before reaching the grasp P_f from a defined distance,
- G_p is a tuple of fingers joint values where the fingers make contact with the object.

Several grasps templates are created offline for each object template using the GraspIt! simulator [58]. A 3D transparent hand or "ghost hand" is projected in the pose relative to the object template. It allows the human operator to visualize the arm configuration needed to grasp the object before actually performing a motion with the real robot. That way, the human operator can choose the location of the hand for a particular task (e.g., in Figure 4.4). The target pose can then be sent to the motion planner (Section 4.6) to generate the required arm trajectories.



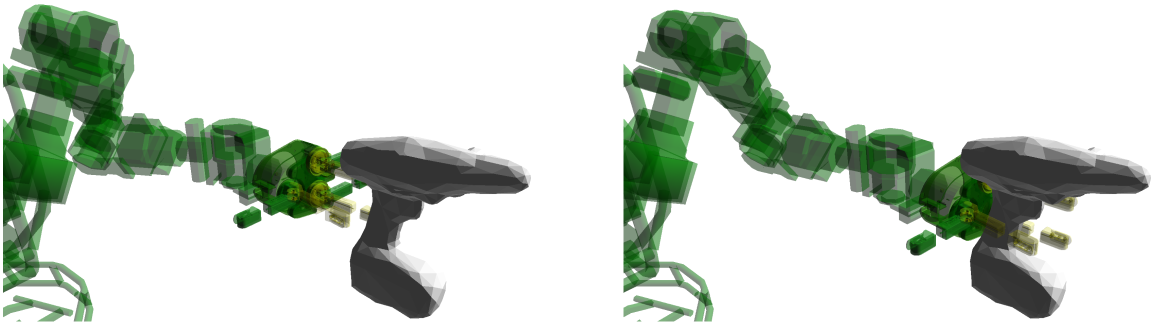
(a) Front. **(b)** 45 degrees. **(c)** Handle.

Figure 4.4: Using a "ghost hand" the final grasps can be visualized (e.g., for the drill template).

Pre-grasps poses P_p are hand poses calculated online to place the hand in the approaching vector V_a near the object. The distance between the pre-grasp pose P_p and the final-grasp P_f has been defined to be around 15 cm away from the object (Figure 4.5a) in order to reduce the risk of collision while reaching the final poses that the end-effector needs to have before closing the fingers around the object (Figure 4.5b).

The *Grasp Template Library XML* definition is standardized based on the *MoveIt! Grasp Message* [61] format and consists of the following tags:

1. *<grasplibrary>*: Is the root of the *XML* file.
2. *<grasps>*: Contains all grasps that belong to that *object template type*.
3. *<grasp>*: Has a unique *ID* and its definition is based on the *MoveIt! Grasp Message*.
4. *<final_pose>*: Pose that the hand will have to reach before closing the fingers.



(a) Pre-grasp. **(b)** Final-grasp.

Figure 4.5: Using a “ghost robot” the pre-grasp and final-grasp poses (here shown for the drill template) can be visualized prior to perform an arm motion on the real robot.

5. <*approaching_vector*>: Pre-grasp pose of the hand, based on vector and distance.
6. <*grasp_posture*>: Joint configuration of the fingers after reaching to the *final pose*.

Listing 4.2: Grasp template for a right robotic hand with two fingers.

```
<grasplibrary>
    <grasps name="fire_hose" type="1" >
        <grasp id="10">
            <final_pose x="0.0" y="0.085" z="0.0" qx="0.7071" qy="0.0"
                qz="-0.7071" qw="0.0"/>
            <approaching_vector x="0.0" y="1.0" z="0.0" desired="0.15"
                minimal="0.05"/>
            <grasp_posture>
                <finger idx="0">
                    <joint name="right_f0_j0" value="1.22"/>
                </finger>
                <finger idx="1">
                    <joint name="right_f1_j0" value="0.0"/>
                </finger>
            </grasp_posture>
        </grasp>
    </grasps>
</grasplibrary>
```

4.3.3 Stand Template Implementation

A stand template contains information about a target pose that defines where and how the robot should be located to be able to reach the pose of an object template. This information is specific to a robot hardware. Each robot hardware is required to define stand poses for each object template. Stand templates are defined by the tuple:

$$s = (N, T, I, P_s),$$

where:

- N = is the name of the object template that the stand pose belongs,
- $T \in \mathbb{N}$ is the type of object template (e.g., tools, debris, hose),
- $I \in \mathbb{N}$ is the ID number of the stand pose,
- $P_s \in \mathbb{R}^2 \times \mathbb{SO}(3)$ defines the pose (2D position and orientation) of the robot pelvis in the ground plane with respect to the object.

Several stand templates are empirically created offline for each object. The pose P_s of the robot pelvis relative to the object is selected in a way that allows the end effector to reach the object with high manipulability. This problem has also been researched in an online autonomous approach as *inverse reachability* [115, 116]. However, for the purposes of this approach, this information is precomputed offline and included as part of the object template. A "ghost robot" can then be projected in the pose relative to the object template. This allows the human operator to visualize the robot pose needed to reach the object before actually performing locomotion with the real robot. That way, the human operator can choose the location of the robot for a particular task (e.g., in Figure 4.6).

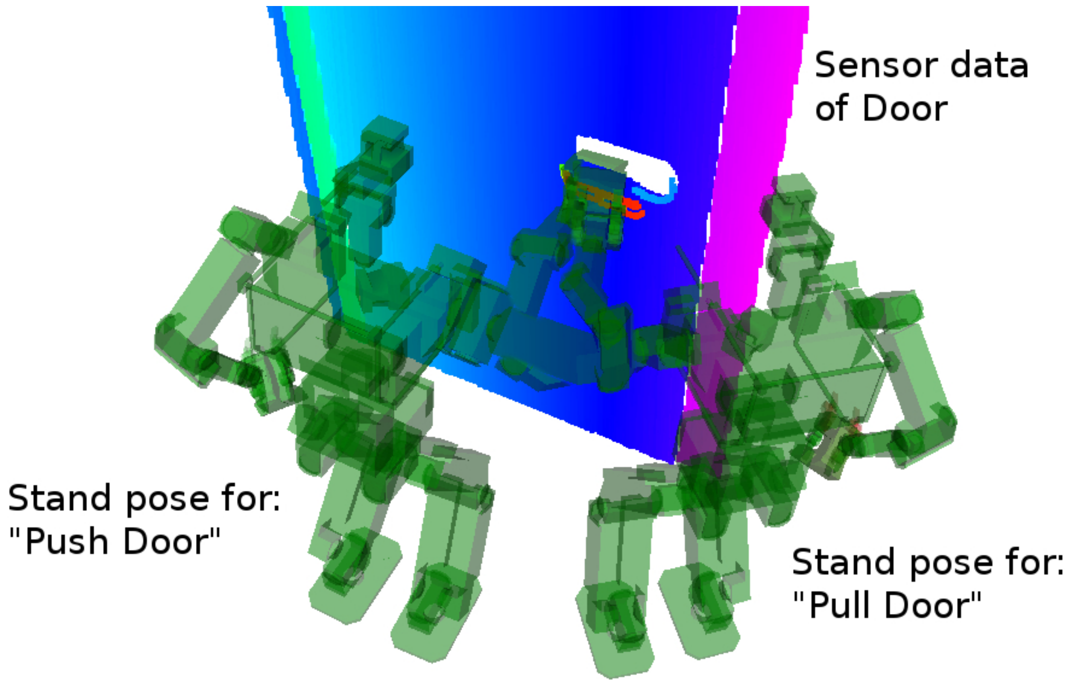


Figure 4.6: Two different stand poses to open a door. For a "Push Door", the robot can stand in front of the door (left ghost robot). For a "Push Door", the robot needs to stand on the side of the door to let the door open (right ghost robot).

The *Stand Template Library XML* definition consists of the following tags:

1. *<standposelibrary>*: Is the root of the *XML* file.
2. *<template>*: All stand poses that provide good reachability for that *object template type*.
3. *<standpose>*: Contains the pose information and has a unique *ID*
4. *<pose>*: The pose of the pelvis with respect to the *object template*.

Listing 4.3: A short version of the stand template for a door.

```
<standposeslibrary>
    <template name="door"  type="8" >
        <standpose id="0" >
            <pose xyz="-0.86 -0.33 -0.12" qx="0.0" qy="0.0" qz="0.7071"
               qw="0.7071"/>
        </standpose>
        <standpose id="1" >
            <pose xyz="0.65 0.50 0.87" qx="0.0" qy="0.0" qz="-0.7071"
               qw="0.7071"/>
        </standpose>
    </template>
</standposeslibrary>
```

After the human operator has selected the appropriate stand pose for the robot, the target pose can then be sent to a locomotion planner. For locomotion planning, 2D grid map slices from regions of interest are created and used to generate a collision-free footstep plan [35]. In [104], a footstep planner approach for humanoid robots is presented (Section 4.6). This footstep planner will generate the required locomotion trajectories to approach to the door.

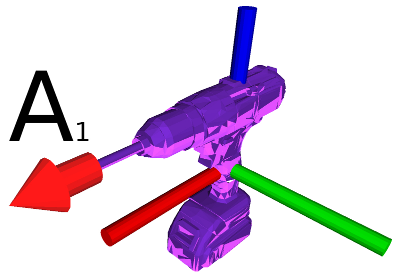
4.4 Affordances in an Object Template

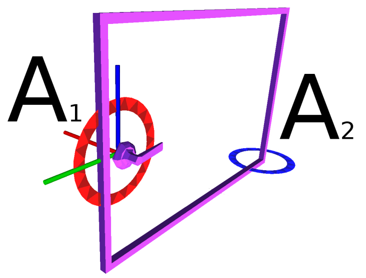
Inspired by the theory of affordances from J.J. Gibson [30] defined actions are created in the object templates which are possible to be performed with the real object. In this approach these actions are defined as constrained translations and rotations in a defined (but not unique) frame of reference of an object template. If the action is a translational motion, the X axis of the frame of reference that describes the affordance needs to point in the direction of the required motion. If the action is a rotational motion, the X axis of the frame of reference that describes the affordance needs to be collinear with the axis of rotation of the required motion. End-effector target waypoints are generated online using the pose of the end-effector as starting pose and the direction of the X axis to generate the final waypoint for a translational motion or the set of waypoints around the axis for a rotational motion. That way, the robot can be commanded to perform actions with an object template at a task level, for example, the valve can be turned using its axis of rotation, the drill can be pushed using the axis of the bit, and the door can be opened by turning the handle using its rotational axis and pushing or pulling the door using the hinge axis. The generation of these constrained path motions of the robot's hand is detailed in Section 4.6.

Of course, objects can have more the one affordance. To define this, each affordance is defined with respect to the frame of reference of the object template. For example, a door will have at least two affordances, one that describes the rotational motion of the handle to unlatch the door and one that describes the rotational motion of the door to open it. Some examples of affordances in object templates such as a drill, a door, and a fire hose can be seen in Table 4.1.

4.5 The Concept of Usabilities in an Object Template

This section describes the concept of *Object Usabilties*. An object usability is a point of interest on an object grasped by the end-effector that can provide a functionality to the agent (e.g., a robot or a human) when executing manipulation motions with respect to another object. Object

Drill Template	$A_1 = \{0, 0, 0, 0, 0, 0, 1.0\}$
	The drill affordance “Insert” is a translation along the X axis (red arrow).

Door Template	$A_1 = \{0, 0, 0, 0, 0, 0, 1.0\}$ and $A_2 = \{0, -0.7, -1.0, 0, 0.7071, 0, 0.7071\}$
	The door affordance “Turn” is a rotation around the X axis (red ring) and the affordance “Open” is rotation around the Z axis (blue ring).

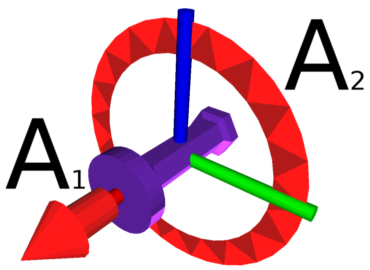
Hose Template	$A_1 = \{0, 0, 0, 0, 0, 0, 1.0\}$ and $A_2 = \{0, 0, 0, 0, 0, 0, 1.0\}$
	The hose affordance “Push” is a translation and the affordance “Turn” is a rotation both over the X axis (red arrow and ring).

Table 4.1: Affordances described in object templates.

usabilities are not necessarily common usable parts of a tool, they can be arbitrary fixed points within the grasped-object frame of reference. For example, if we consider a golf stick, the common usable part would be the head when executing the “hittable” affordance of a golf ball, however, a usability can also be defined at any arbitrary point along the stick.

4.5.1 Definition

In contrast to affordances, where the possible actions to be executed with an object are described, object usabilities provide the reference point on an object grasped when executing a manipulation motion described by the possible actions of another object. Object usabilities do not describe how the object should be manipulated, they describe which part of the object should be taken into account when manipulating such object. Affordances and object usabilities are strictly related since object usabilities define the frame of reference located on the object grasped by the end-effector which is going to be used to execute an affordance of another object.

The formal definition of an object usability is as follows:

$$u = {}^{ot}P \in \mathbb{R}^3 \tag{4.1}$$

where the object usability u is defined as a three dimensional point P described in the frame of reference of the object template ot.

In principle, usabilities can also be located in the end-effector, for example, to describe which part of the end-effector (finger, palm, or other side) is going to be used for a manipulation task, but this paper will focus on usabilities located on objects grasped.

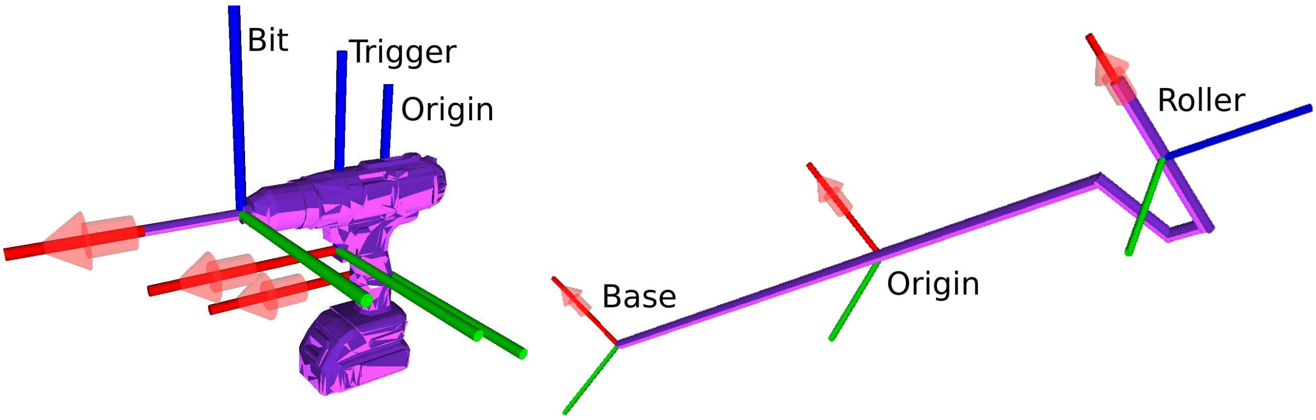


Figure 4.7: Object Usabilities. The Drill Template (left) has three usabilities: Origin, Trigger, and Bit. The Paint Roller Template (right) has three usabilities: Origin, Base, and Roller.

Including the object usabilities in the OT, allows the operator to select points of interest in a grasped object on the fly so that this points can be used while planning and executing motions. Instead of having one "tool tip" per object, the OTL can include descriptions of multiple points in the reference frame of an object. For example, as previously shown in Figure 4.7, the Drill Template will have at least three usabilities: the origin of the template, the ON-OFF switch (trigger), and the bit. Usabilities allow objects that are grasped by the robot to be considered as online-augmented end-effectors since with this information, affordances of other objects can then be executed using these points as reference for motion planning. The "bit" in the drill is located around 10 cm above the origin of the reference frame of the Drill Template, for this reason special planning has to be done to achieve the desired cut pattern in the wall (see Figure 4.8).

4.5.2 Implementation

To implement the usabilities in the OTL, the <*usability*> tag was added and it follows the same syntax used to define frames of reference in the *Unified Robotic Description Format* that is used to describe the kinematic structure of a robot [106]. This tag identifies usabilities by a number, includes a name description, and a pose in the frame of reference of the object. The pose that defines the location of the usability in an object template uses an *xyz* argument for position information and four arguments to represent the rotation in the form of a quaternion *qx*, *qy*, *qz*, and *qw*. A short version of the *XML* definition for the Drill Template can be seen in Listing 4.1 :

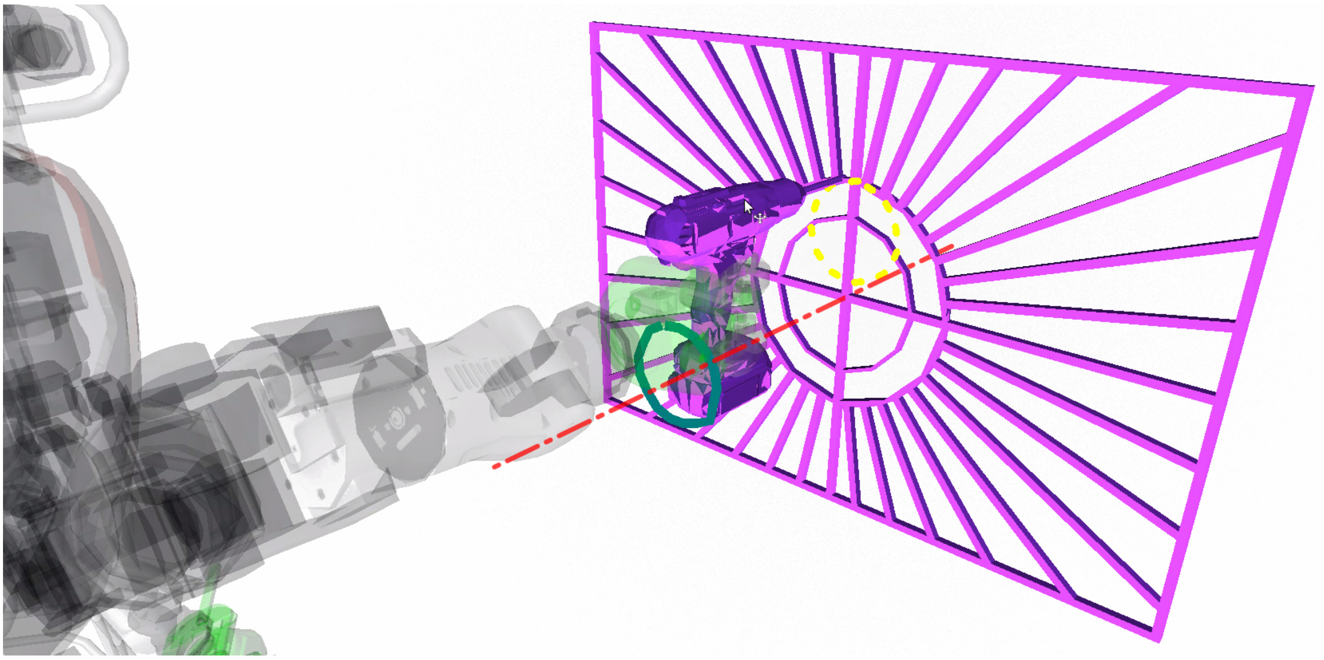


Figure 4.8: Robot's end-effector path for cutting a circle in the wall using a "Gun Drill" (green circle) rotates around the Wall template axis (doted red line). Without selecting any point of reference in the drill for motion planning, the cut pattern will not be correct as shown by the yellow doted line.

Object usabilities are considered to be fixed in the frame of reference of the object; in the context of this approach non-rigid objects are not considered. To be able to create motion plans with respect to the usabilities in the object, the object template needs to be "attached" to the end-effector. The human operator can manually align the object template or assist an autonomous algorithm to match the pose of the real object once it has been grasped by the robot. This attachment creates a virtual rigid link between the end-effector and the object, thus, augmenting the end-effector with the object. Although motion plans are created with the assumption that the grasp is rigid, this is not the case due to real world imperfections. The human operator can request at any point that the object template gets "detached" from the end-effector, then, after matching the new pose of the object, the object template can be reattached. The final transformation between the usability u defined in Equation 4.1 and the root of the kinematic chain can be obtain by a rigid body transformation as seen in Equation 4.2.

$$^{root}T_{ee} = {}^{root}T_{ee}\, {}^{ee}T_{ot}\, {}^{ot}P \tag{4.2}$$

where:

- $^{root}T_{ee} \in \mathbb{R}^3 \times \mathbb{SO}(3)$ is the transformation between the $root$ of the kinematic chain and the end-effector ee.
- $^{ee}T_{ot} \in \mathbb{R}^3 \times \mathbb{SO}(3)$ is the transformation between the end-effector ee and the object template ot. This transformation is manually found when the operator aligns the object template with the real object.

Continuing with the Drill Template example, Figure 4.9 shows the frames of reference of the kinematic chain of the right arm of the robot after the attachment has been done. To generate linear translational motions of the desired usability, no additional transformations are required since both, the end-effector of the robot and the object usability will move with respect to the same path. However, to generate circular motions with respect to a specific axis of rotation, special planning has to be done since the “bit” in the drill is located nine centimetres above the origin of the reference frame of the Drill Template (where nominal end-effector poses are designed to grasp drill).

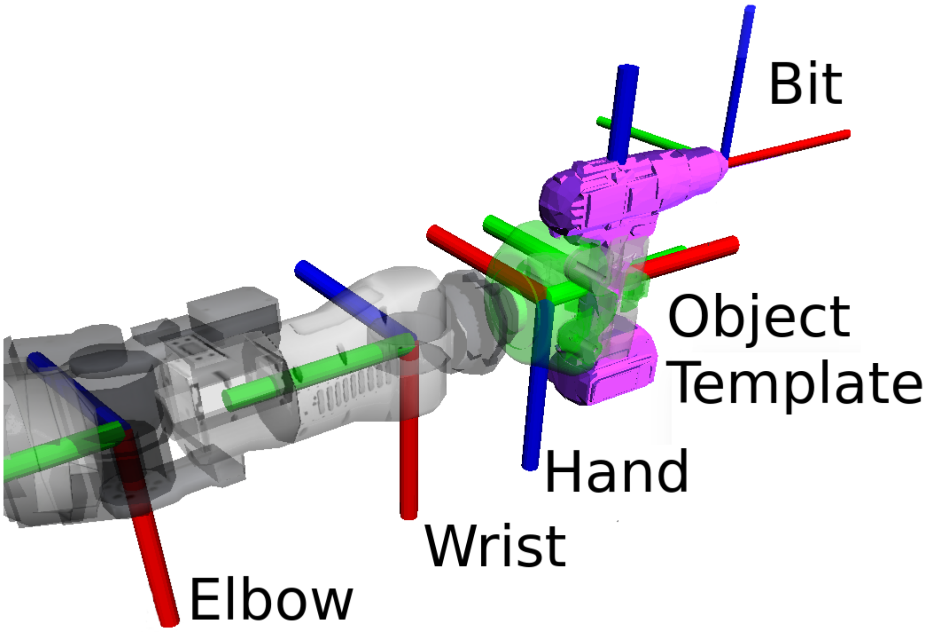


Figure 4.9: Frame of reference of the arm joints considering the bit in the Drill Template as the frame of reference for motion planning.

To generate a circular path of the drill, the circular affordance of the Wall Template can be used as an example. Figure 4.10 shows a setup to perform a circular cut in a wall using the affordance motion from the Wall Template and taking into account the “bit” usability of the Drill Template as reference point for planning. To perform the correct path, the usability axis shown in blue color needs to rotate around the affordance axis of the Wall template which is shown in red color. However, in the backend motion planning that will be described in Section 4.6, all motion plans still need to be performed with respect to the robot’s hand. For this reason, a virtual axis (shown in light green) needs to be created so it can be used as rotational axis for the robot’s hand. If no usability was to be considered, the final path of the end-effector will not generate the correct motion path (small circle in dark green), since the radius that would be considered will use the distance between the robot’s end-effector (shown in black) and the Wall axis (shown in red). The virtual axis is calculated subtracting from the robot’s hand the vector between the object usability and the affordance axis of rotation. Finally, the robot’s end-effector axis (black) rotates around the virtual axis of rotation (green) generating the correct path for the robot’s hand, shown as the big circle in dark green.

4.6 Manipulation Implementation in an Object Template

Executing manipulation motions using objects requires planning collision free paths between start and goal joint configurations and for generating joint trajectories based on cartesian end-

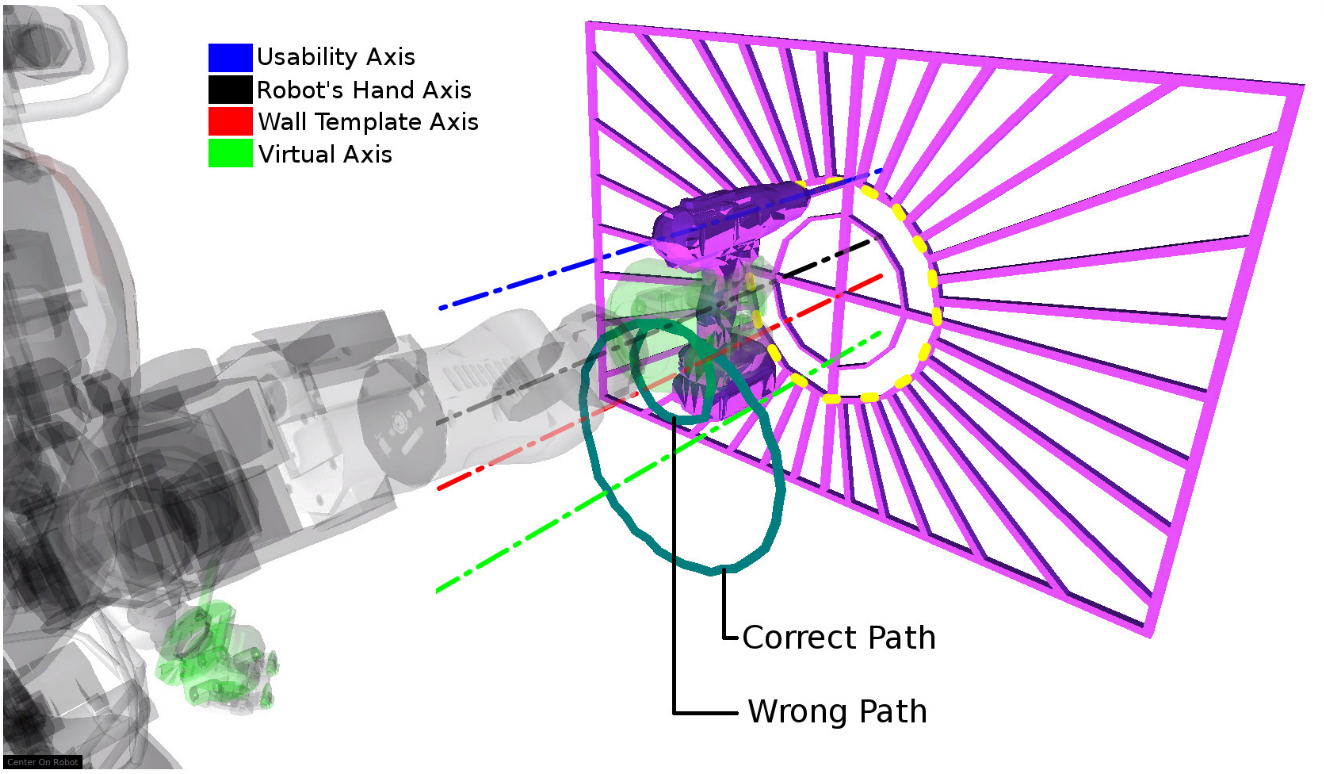


Figure 4.10: Transformations to generate the correct path of the end effector when planning with respect to an object usability. Doted lines show the respective axis of the frames: Usability in blue, Robot's hand in black, Wall Template circular affordance axis in red, and the generated virtual axis where the robot's hand should rotate around is shown in light green. Two circles in dark green represent the paths of the end-effector, the smaller circle represents the path when no usabilities are used and the bigger circle represents the path when the object usabilities are used.

effector trajectory requests. The default motion planning backend used in this approach is based on the MoveIt! [15] planning framework and extends the standard ROS Action interface for planning requests provided by it.

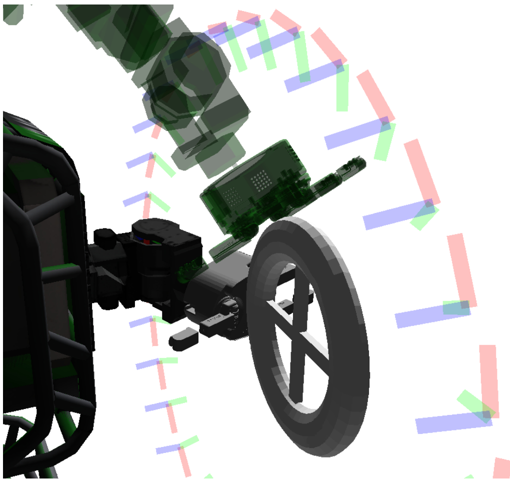
For simplicity, in this approach, only linear and circular motions of the end-effector are considered. Internally, all cartesian motions are represented by short linear segments that are sampled using an iterative inverse kinematics solver that always takes the preceding state as an input. To perform a circular motion, the rotation axis, direction, and rotation angle have to be specified. Afterwards, several linear segments are used to follow the circular motion which greatly simplifies commands for complex motions. Using this approach, smooth trajectories can be generated quickly and reliably.

A planning reference pose with respect to the end-effector frame can be specified as part of the motion planning request [43]. Thus, the planner interface allows for planning with regards to arbitrary provided reference points, which is a capability that is used for the remainder of this work to generate motion plans with respect to object usabilities.

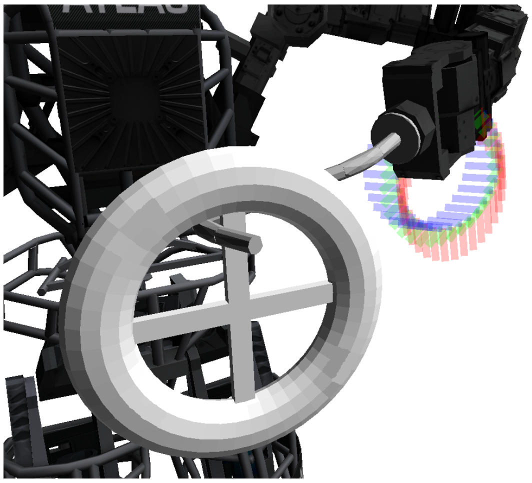
As an alternative to the MoveIt! based backend, the Drake planner as described in [23] can optionally be used. It allows for planning whole body motions, but does not support collision avoidance based on a octomap generated from sensor data.

4.6.1 Constrained Motion Planning

The affordances defined in each template are used to create motions that are constrained to follow a Cartesian path between the initial and final end effector's pose. Waypoints are generated based on linear interpolation between initial and final poses. By using Spherical Linear Interpolation (slerp) [98] orientations for the end effector's goal pose can be different from the start end effector's orientation. More complex constrained motions such as circular motion are generated by concatenating multiple short linearly interpolated Cartesian paths. Additionally, this basic motions can be combined to generate spiral motions. By considering a pitch value, each circular waypoint can be shifted along the axis of rotation to generate a spiral path of waypoints. These constrained motions can also be designed to maintain the end effector's orientation as shown in Figure 4.11 [81].



(a) Rotating valve with hand.



(b) Rotating valve with hook.

Figure 4.11: Circular path plan to turn a valve 360 degrees. In (a) the hand rotates around changing its orientation, while in (b) the hook rotates around keeping its orientation. Interpolated poses are shown for the last joint of the arm.

To avoid collisions with the environment [52], the robot creates a 3D Octomap [36] representation of the world model, this will be described in Section 5.3.2.

4.6.2 End-effectors and Grasp Agnostic Manipulation

On the one hand, to increase the efficiency and robustness of manipulation tasks, the end-effectors need to meet some requirements. On the other hand, the manipulation approach needs to account for the specific abilities of different types of hands. End-effectors can have actuated, under-actuated, or fixed joints as well as sensors to provide feedback to the supervisor. Like in humans, having actuated opposing fingers is key for holding objects. The number of DOF and finger configuration allows the robot to have more ways of grasping (having control over) an object. Then, if available, tactile feedback, force sensing, and image information allows the operator to gain more awareness of the quality of a grasp or a motion. For example, with more advance grasping capabilities, more different ways of manipulation motions can be executed. When performing manipulation tasks in environments designed for humans, there is a high possibility of requiring human-like grasping of objects. The current implementation of this approach allows the use of a wide breadth of this capabilities by considering any number of DOF to store potential grasp postures for position-controlled end-effectors. However, it is possible to extend this approach to consider additional grasping capabilities. For example, force-controlled grasps or tactile sensing distribution information can be included in the grasp library.

Manipulation planning should be robust to inaccuracies within grasping that can happen while interacting with objects in an uncontrolled environment. For this reason, in this approach, each time an affordance is requested to generate manipulation motions with an object, instead of pre-defining target goals for the end-effector, the current pose of the end-effector is considered as starting pose to generate constrained motions. This gives the advantage with respect to other state-of-the-art approaches, that generation of manipulation motions does not depend on the end-effector capabilities or the grasp pose. This way, if robots are able to physically interact with the object by grasping it or by using its end-effector to move the object (assuming that this interaction is not blocking the task itself) the online generation of motions is created considering the current pose of the end-effector. For example, as previously seen in Figure 4.11a, the robot can grasp the valve sideways, but it can also grasp the valve from the front side, and the motion planner will still generate the required trajectory to make the end-effector rotate around the valve.

4.6.3 Planning with Grasped Objects

Based on the capability of MoveIt! to attach and detach collision objects, a similar behavior with object templates has been implemented. Each object template is analogously created as a collision object in the MoveIt! Planning Scene [62]. Object templates can be attached to the robot's end-effector, this way, when the robot is commanded to execute a motion with the end-effector, the object template will keep its pose with respect to the end-effector (this is assuming a rigid grasp). After an object template has been attached to the end-effector, a collision object in the planning scene is simultaneously attached to the link of the robot that corresponds to the end-effector. With this attached collision object, motion planning can be generated free of collision with the perceived environment while manipulating an object template.

4.7 Allowing Versatility and Robustness to Solve Manipulation Tasks

The work presented in this Section was published in: 2015 15th IEEE-RAS International Conference on Humanoids Robots [77].

This section describes the concept of versatility in the context of robotic manipulation using object affordances to allow a human operator to improvise manipulation tasks. Improvisation is a powerful human ability that has not yet been deeply explored in robotic manipulation research. An interesting research approach by Stilman et al. presented the "MacGyver" paradigm as a research problem [100]. They propose that robots should be able to use arbitrary objects in the environment to solve unforeseen manipulation tasks. Performing these actions autonomously for unknown tasks in real world scenarios with degraded conditions is not feasible within the next years. Some autonomous approaches like [105], [112], and [114] have demonstrated autonomous capability for obtaining functionality information from objects. However, these approaches still require development to be able to perform autonomously in less controlled environments and unforeseen tasks. The idea of a robot applying manipulation skills to different objects has also been suggested by Leidner et al. [53]. In a semi-autonomous remote supervised approach, a human operator can effectively aid the robot to apply the necessary manipulation skills needed to achieve the task.

Inspired by the work of [100] this thesis proposes a “MacGyver” paradigm but for supervisor-assisted humanoid robots. In this paradigm, versatile manipulation is the key ability to succeed in a particular task where expected known objects are not present. Versatile manipulation using the proposed approach allows the human supervisor to utilize skills designed for previously known objects for new unknown objects on the fly, utilizing the object template of a similar known object and provided that objects found in the environment have similar properties to previously known objects. In the context of this approach versatile manipulation refers to the different ways on how tasks can be achieve by moving an object through the environment. This approach focuses on defining affordances for some objects to achieve a task, and how a human operator either apply these affordances as designed, or utilizes them in a newly way which was initially not planned.

As concluded by Liu et al. [55] in humans, grasps are distinguished by features related to the grasping action such as the intended motion, force, and stiffness. The ability of humans to transfer these properties between objects increases the rate of success in manipulation tasks. In a similarly way, these properties are needed for robot control and the ability to transfer them between similar objects can help to achieve manipulation tasks. Doing this in a full autonomous way in unstructured and degraded environments is not feasible within the few next years. For this reason, the assistance of a human operator can help in identifying the tasks requirements and the objects that could be used to achieve them.

For example, consider the case where the robot enters a degraded environment and the human supervisor identifies the next task objective is breaking a glass panel to gain access to a fire hose or to allow smoke to dissipate. A commonly used object for this task would be a hammer. However, if a hammer is not available in the environment, similar objects like debris, pipes or other tools that have similar properties to a hammer could be utilized with the same manipulation motions designed for using a hammer.

4.7.1 Object Classification

During a disaster scenario multiple objects from a wide variety of types can be found in the environment. In this approach, the object-space that the robot can use is constrained to objects that can be grasp and manipulated with a hand. To make clear the object-space that is considered in this thesis, a classification in two groups is given:

Floating Objects

Refers to the objects mobile in the environment that can be grasped and lifted. For this reason similar physical properties between objects are needed to transfer affordances. Size, mass, center of mass, hardness among others need to be similar in such a way that the task might still be possible to achieve. In the hammering example, objects that would fit in the hand and that have the hardness and mass necessary to break a window could be used instead of the hammer as shown in Figure 4.12.

(a) Hammer

(b) Metal Bar

(c) Bolt

(d) Vertical Drill

Figure 4.12: Floating objects. Mass, center of mass, hardness and size properties are similar enough to use them to create a force impact into another object.

Constrained Objects

Refers to the objects that have limited degrees of freedom in the environment. In this case, physical properties of the objects are not so relevant compared to the motion constraints that define the use of the object. For example, the drawer of a desk, a door handle and the door itself have motion constraints that can be defined as linear or circular motions with respect to an axis in a frame of reference as shown in Figure 4.13.

4.7.2 Manipulation Classification

This section describes the classification of manipulation tasks that is considered in this approach. To be able to transfer manipulation skills between two similar objects it is important to define in detail which properties of each manipulation motion are needed to accomplish a task. This approach is constrained to a representative set of manipulation tasks based on the classification of human manipulation behaviour made by Bullock et al. in [12] as seen in Figure 4.14.

Based on the manipulation properties described by [60] and [25] six classes are selected which are identified to be most commonly used in manipulation tasks. The nomenclature used to identify each class represents the constraints for the six DOF. This representation uses four letters, each of them represents a type of motion in a dimension. The letters used are “u” for unconstrained motion, “t” for a motion that is only translational, “r” for a motion that is only rotational, and “x” to represent a fixed constraint. The selected classes can be seen in Table 4.2.

(a) Door and Handle **(b)** Sliding Door **(c)** Drawer

Figure 4.13: Constrained objects. Degrees of freedom are well defined, the door and handle have one axis of rotation, the drawer and the sliding door have one axis of translation.

Class	Description	Example
uuu	Free Motion	Moving a floating object
uur	Point on a plane	Drawing on a white board
ttr	Surface against surface	Cutting with a vertical drill
uxx	Cylinder in slot	Attaching a hose to a pipe
rxx	One rotational DoF	Turning a doorknob
txx	One translational DoF	Pulling a drawer

Table 4.2: Manipulation classes to be used in this approach. Table based on [25].

4.7.3 Manipulation Transferring Types

In this approach three different types of examples to differentiate the ways in which a manipulation task can be achieved have been defined. Each one of these example types represent a way of transferring manipulation skills; they are described as follows:

Manipulation Transfer Type 1

Transferring manipulation skills between objects in which physical properties can differ, but that they can still be considered to be the same object. For example, turning valves of different radius or with different number of cross bars, pulling drawers of different shapes and lengths. This is a simple way of transferring skills between objects, and the cases where objects are exactly the same is considered as a special case of this type of manipulation transferring.

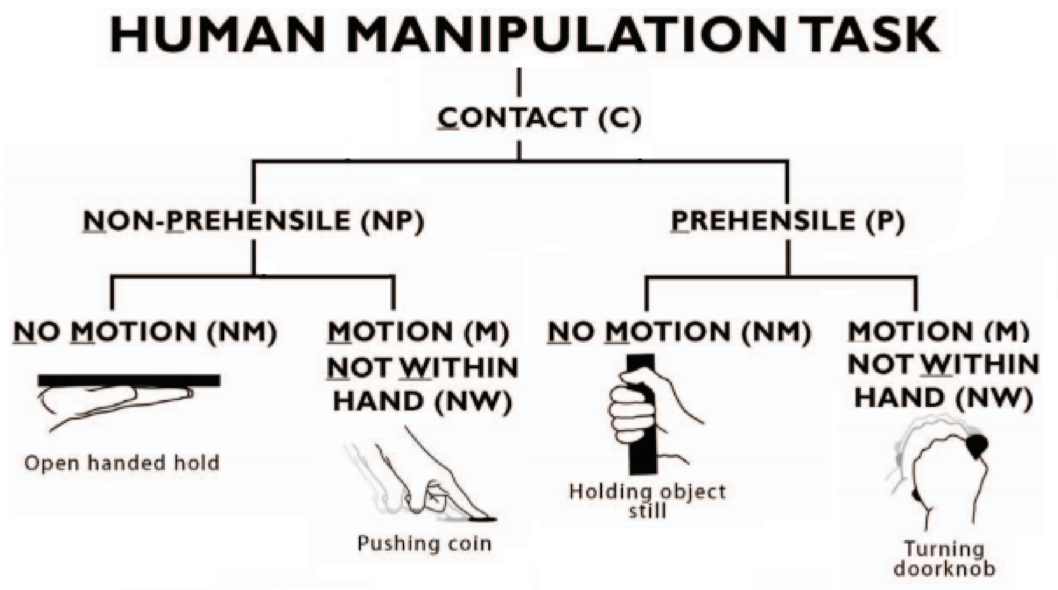


Figure 4.14: Manipulation classification to be used in this approach marked inside red line. Image based on [12].

Manipulation Transfer Type 2

Transferring manipulation skills between different objects, but with same manipulation classes. This means that the affordances of the objects can be utilized in the same way. For example, a common doorknob manipulation class is “rxx” which allows rotation in only one axis, this means we can use the turn affordance of the valve template which belongs to the same manipulation class. Pushing a box under a table requires a manipulation class “txx” which can be achieved by using the push affordance from a drawer. This type of manipulation transferring shows how a robot is capable of using an object based on manipulation skills designed for another object in a way that will allow the robot to achieve the task goal.

Manipulation Transfer Type 3

Transferring manipulation skills through the use of intermediate objects. This type of manipulation skill transferring can be seen as the “MacGyver” paradigm described in this approach. This refers to how the robot is capable of manipulating an object for a different purpose than the one it was designed for and potentially utilizing manipulation skills from another object. The improvisation ability is still provided by the human supervisor, but it is now possible for the human supervisor to transfer manipulation skills from previously known object allowing the robot to use new unknown objects. This increases the potential of the human-robot team to continue the manipulation task.

5 Human-Robot Interaction Through Object Templates

A condensed version of Sections 5.5, 5.6, and 5.7 has been conditionally accepted for publication in: Journal of Field Robotics, Special Issue on the DARPA Robotics Challenge Finals [79].

In this chapter, the interaction between a human supervisor and an avatar robot through the concept of object templates is presented. The concepts outlined in Chapter 4 show how information of the manipulation of objects is stored in a single entity of information, which at the same time can be used by the human supervisor and the remote robot.

This interaction presents advantages to the existing approaches for remote semi-autonomous robots. These allow a human supervisor to interact with a remote robot at an affordance level. They also allow a human operator to select the point of interest in an object which is going to be used for planning. This means that known objects in the environment can be used as online-augmented end-effectors. As a further contribution of these concepts, the human operator can apply the manipulation skills from an object template as presented in Chapter 4 to another object which may not necessarily match the object template used. Utilizing object templates to other purposes than the one they were originally designed for, allows a human operator to increase the robustness in the achievement of a manipulation task, for example, by improvising [100].

To evaluate these concepts, a human operator can utilize objects in the environment using selected points of interest from these objects. The robot must be able to grasp and manipulate objects in the environment in order to “attach” them to the kinematic chain of the robot’s arm. The operator has the possibility to adjust the object template to match the pose of the real object as grasped by the robot’s end-effector. This online-adjustment of end-effectors accounts for grasping inaccuracies providing more robustness and efficiency to achieve the overall manipulation task.

5.1 Related Work

Affordance Template ROS Package

In Section 3.1 this approach was discussed as a concept for manipulation control. Here the Graphic User Interface (GUI) aspects of this approach are described. This package is implemented as a plugin for the tool interface “Rviz” which is software used to visualize in a virtual environment robot information such as *joint states*, *image data*, and *3D perception data* among others. Using this package a human operator can iterate through the predefined end-effector waypoints and also execute them in reverse mode to account for errors during grasping.

Action Templates

In [7], a shared autonomy approach to provide a remote robot with object information through a GUI is presented. This GUI is created as a tablet computer application to provide an operator with information of the environment acquired by the robot sensors. Compressed video and compressed action descriptions are sent to the robot tolerating latency. CAD models of the object of interest are rendered on top of the video stream to provide a “ring” menu of the possible actions of the object.

Tool Affordances

Tool affordances have been previously used for autonomous robots, however, these approaches require the use of well defined objects, colors, and backgrounds (Section 2.2.2), which are unlikely to be present in disaster scenarios or unstructured environments. For this reason, the use of a human operator in the loop that by the presented concept has the ability to select in a GUI the affordances and usabilities of the objects of interest presents advantages to the existing state-of-the-art autonomous approaches.

5.2 Contribution

The contribution in this chapter is an approach for human-robot interaction at an affordance level with a remote robot using a GUI. This approach allows a human operator to command execution of manipulation tasks using the grasps, affordances, and usabilities aspects described in Chapter 4. The use of this GUI gives a human operator the ability to specify how objects should be manipulated by providing an interface to select the affordances required to perform a task and has been demonstrated for challenging real-world scenarios. Also, the operator has the ability to online-augment the robot’s end-effector by “attaching” the object templates to the kinematic chain, providing information to the robot about the object’s pose described in the end-effector frame of reference. Related publications are [44], [45], [76], [77], and [46].

Multiple people have contributed to the GUI that the contribution within this thesis is part of. These contributors are named here: Stefan Kohlbrecher (worldmodel), David C. Conner, Ben Waxler, and Shawn Hanna (control, communication), Alexander Stumpf (Footstep planning) Philipp Schillinger and Spyros Maniatopoulos (Behavior control), Felipe Bacim and Brian Wright (OCS), and the author contributed with the affordance-level, grasping, and object control implementation in the interface described in this chapter.

5.3 Operator Control Station

To interact with a remote robot, a human operator requires knowledge of the state of the robot, the state of the environment, and also be able to command the robot. For this reason, an OCS is required, which will additionally serve as means of visualization of a virtual representation of the information acquired by the sensors of the robot (see Figure 5.1).

5.3.1 Requirements

In order to supervise remote robots, human operators require perception of objects in the environment and aid the robot providing this information. In the OCS, a human operator can interact with a remote robot at different levels. On the low level, the human operator should be able to request joint motions. At a middle level, the human operator should be able to request arm motions in Cartesian space. However, for efficient object manipulation, the level of commands that a human operator sends to a robot should be at a task-level, provided that the robot can autonomously execute commands at an action-level.

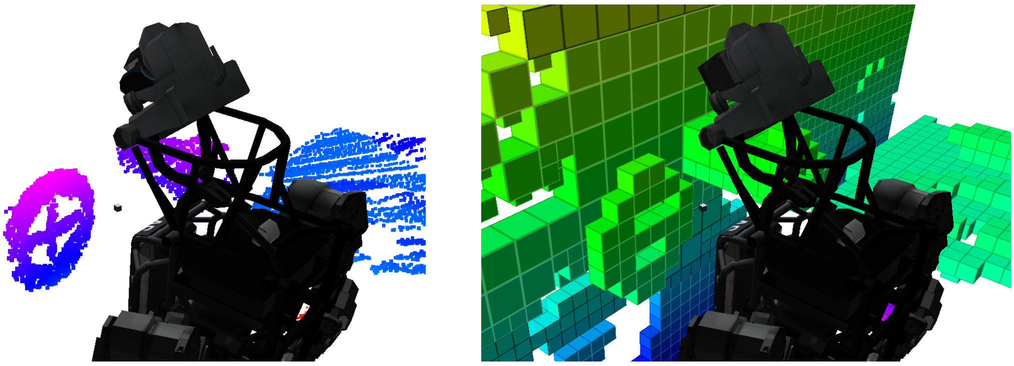
The object template concept presented in Chapter 3 can be use in an OCS where the 3D geometry mesh of the object templates can be visualized and manipulated. However, the OCS has to provide the human operator with *situational awareness* for allowing the human operator to make sense of the remote environment thus allowing a proper decision of the tasks the robot is required to perform. The term Situation Awareness (SA), was defined by Endsley as "*the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future*" [22][119]. Additionally, to provide an insight of the environment as well as safety to the robot system, a model of the world is required such that the human operator can prevent dangerous situations and the robot can generate motion plans free of collisions.

Figure 5.1: Operator control station.

5.3.2 World Model Generation through Robots Sensors

A 3D model of the environment is generated by aggregating sensor data from Light Detection and Ranging (LIDAR) and cameras. This worldmodel is made available to both the robot and the human operator through the OCS using a *world model server* [43]. The sensor data from the LIDAR is visualized in the OCS using open source libraries like the Point Cloud Library (PCL) [89] and octomap [36]. The world model of the environment is required by the robot to be able to generate manipulation motions that avoid collisions with the environment. Figure 5.2 shows an example of world modelling using both aggregated LIDAR data and an octomap representation of the same data.

Additionally, a pose estimate of the robot is obtained by the IMU on its pelvis, and track of different coordinate frames is kept in order to fully reconstruct the pointclouds requested relative to different fixed frames. Using robot pose estimation, internal joint sensing, and external sensors, a robot model that can also be visualized in the OCS as seen in Figure 5.2a is generated. This model is used for multiple applications such as visualizing all joint states, self filtering from sensor data, and collision avoidance.



(a) World and Robot model. **(b)** 3D Octomap.

Figure 5.2: World model of a valve scenario with robot model and the 3D Octomap for planning.

5.4 Integration of Object Templates into an Operator Control Station

Once the human operator has gained sufficient SA and has an available worldmodel, the next step is to start planning manipulation using object templates. As described in Chapter 4, object templates can be represented as a 3D geometry mesh that can be simultaneously visualized with the world model. The human operator aid the remote robot with perception of objects in the environment using the sensor data provided by the robot sensors. Using for example a point cloud from the LIDAR, an operator can identify objects of interest, and using the OCS insert in the virtual environment the object template that corresponds to the object of interest. The human operator is able to change the pose of an object template by using the *Rviz interactive markers* [85] to translate and rotate the 3D geometry mesh and overlap it with the sensor data (see Figure 5.3).

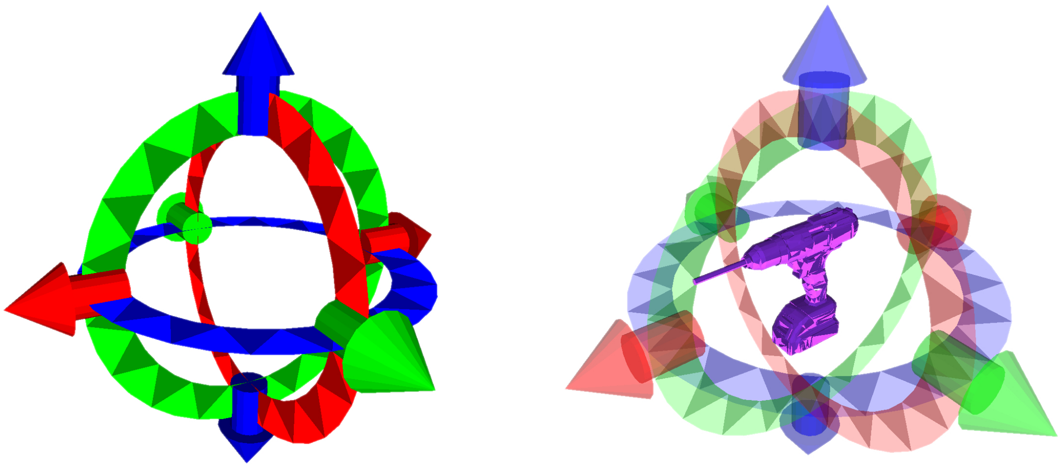


Figure 5.3: Interactive marker (left). Drill template with semi-transparent interactive marker (right).

The user interface used to interact with the remote robots consist of a manipulation widget for each end-effector (see Figure 5.4). This widget is responsible of providing to the human operator as well as to the robot all the functionalities that the OT approach has.

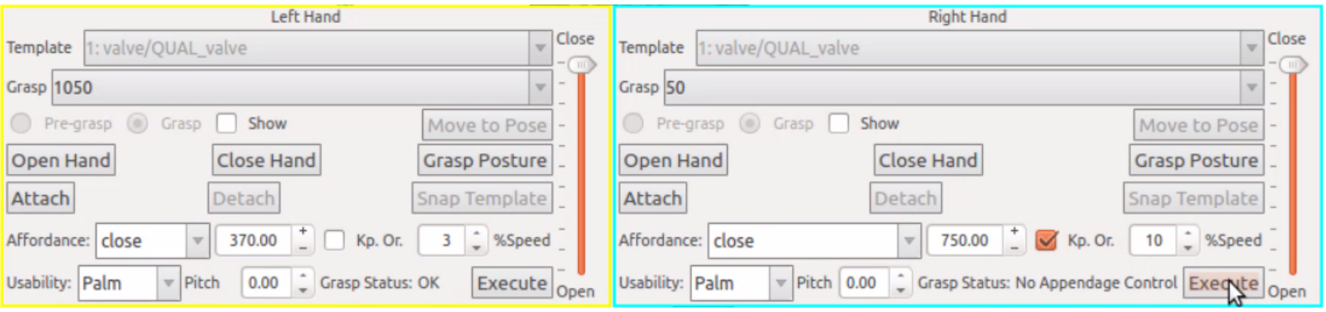


Figure 5.4: Manipulation widgets for each hand.

Once an OT is inserted in the environment, the operator can double click that OT to let know the Manipulation Widget that that is the OT of interest. The Manipulation Widget then displays all the information available for this OT (see Figure 5.5). Pre-grasp and final grasp poses for a specific Grasp Template can be shown. The fingers can be opened, closed, set to the specific joint configuration defined for that Grasp, and also there is the possibility to select the percentage of closure if the fingers are going to be manually controlled. If the object is going to be moved around the environment, the operator can "Attach" the OT to the robot, allowing the motion planner to consider the real object for collision avoidance, in the same way the OT can be detached from the robot. The Usability combo box allows the operator to select the frame of reference in the end-effector that the motion planning is going to be done with respect to (e.g., Palm, Poke Stick, the origin of the template, or any point of interest included as a usability in the OTL). Affordances can be executed with different parameters. Once the affordance is selected from the combo box, the default values for that affordance are automatically loaded, afterwards the operator can change these parameters. The displacement parameters uses degrees for rotational motions and meters for translational motions. The operator can also select if the motion is going to be performed keeping the end-effector orientation or not. In case the affordance is rotational, the operator can give a pitch to that affordance to convert the circular motion into a "spiral" motion. Finally, the speed of the motion execution can also be set.

5.4.1 Communication Constraints

Remote communication with robots navigating inside an environment that has been degraded can be severely challenging. This challenges include low bandwidth, connection interruptions, and latency. Teleoperated approaches can tackle this challenges by using cables for communication, however this presents disadvantages such as limiting mobility and the risk of entanglement. For this reason, a semi-autonomous approach needs to account for situations where communications are limited.

From one side, communications to the robot need to be minimized, thus the commands sent need to be at a higher level than joint teleoperation in order that the robot can use this command and continue with mission execution in an autonomously way if connections are lost. From the other side, robot feedback such as LIDAR pointclouds and camera images need to be compressed to provide sufficient situational awareness to operators even in low latency conditions. Sensor data compression for feedback to the operators was performed using a communications bridge between the OCS and the robot, for more details on this topic see [45].

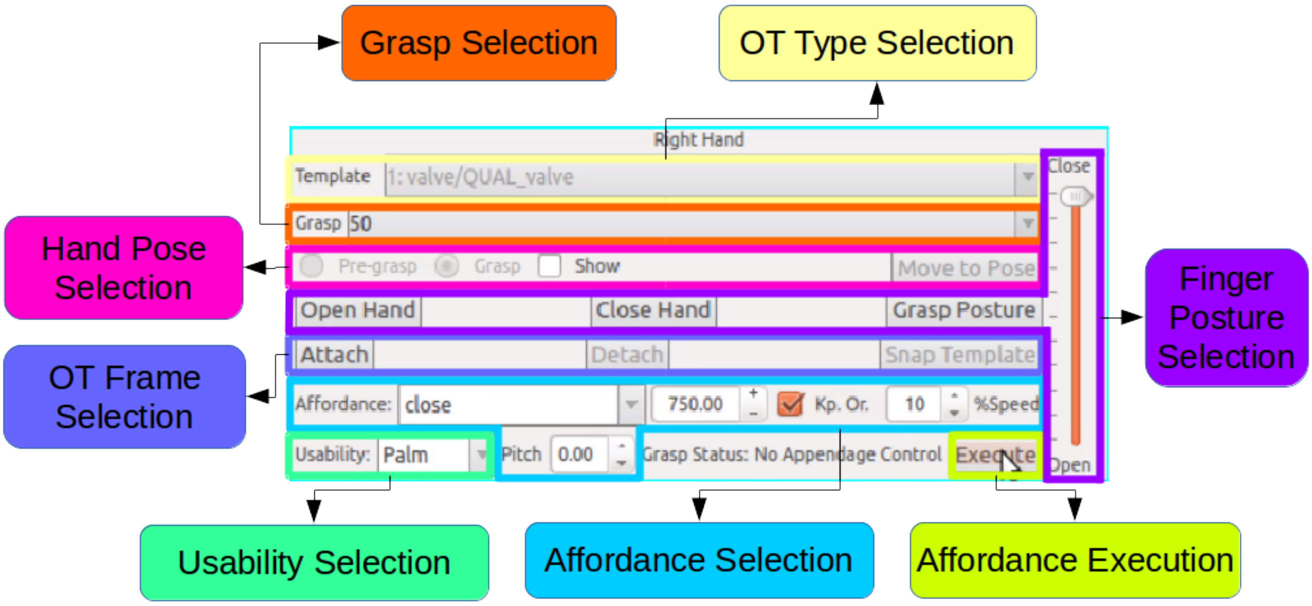


Figure 5.5: Description of the functions in the manipulation widget to interact with object templates.

5.4.2 Object Template Server

Object template's information is provided to the robotic system through the Object Template Server (OTS). The OTS is responsible of loading and providing OT information to any client that requests it. For example, the Main View widget will request 3D geometry mesh information from the object template to display, as well as finger joint configuration while displaying potential end-effector poses to grasp such object. Other clients such as the Manipulation Widget (Figure 5.5) could request grasp information and affordance information from the OTS. Additionally, it will be described in Section 5.7 that the OTS provided information to be used by autonomous behaviors.

Given the possible network constraints that a disaster environment can present, the OTS is required to provide information for both, the OCS side and the Onboard side. On the OCS side, the OTS provides information to all the widgets that use OTs. It also manages the instantiated OT that the operator has inserted in the 3D environment. To replicate the same status in the Onboard side, another instance of the OTS is created in the Onboard side. The OTS in the Onboard side is responsible of keeping OT information to be considered for motion planning, e.g., as collision objects or attached collision objects to the robot. Both OTS were kept synchronized through the Communications Bridge. In case there was any synchronization issue, both OTS were able to be re-synchronized by resetting the instantiated OT information. The architecture of the OTS can be seen in Figure 5.6.

5.5 General Manipulation Task Pipeline

The workflow of subtasks in an object template based approach requires that the operator identifies the sensor data in the OCS that corresponds to the real object. After object identification has

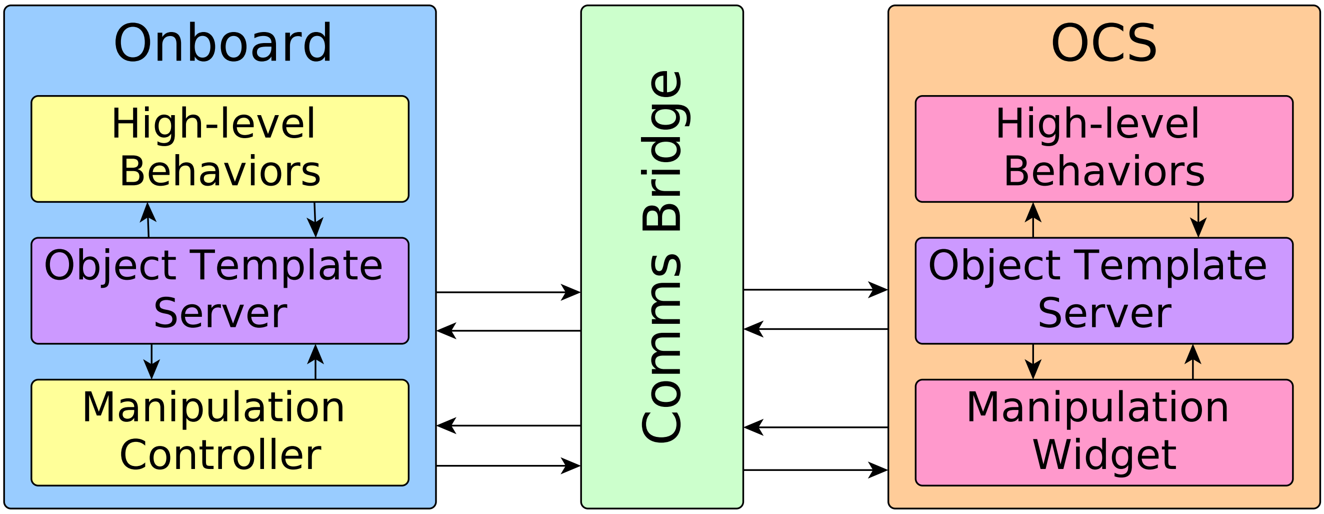


Figure 5.6: Object template server (purple) is instantiated in both, OCS (orange) and onboard (blue) sides. Each OTS provides information to the controller blocks in onboard (yellow) and to the user interface widgets in the OCS (pink). Additionally, both OTS are kept synchronized through the communications bridge (green).

been performed, the following steps are required to perform manipulation with object templates (see Figure 5.7).

1. The operator inserts the template designed for the object of interest using the OCS. The operator manually aligns the 3D model of the template to sensor data that corresponds to the real object. In the current state of this approach, this step is the only one that is performed exclusively by the human operator. Any of the following steps can be executed either by a human operator or by any high-level behavior.

2. With the template in place, a stand pose of the robot close to the current pose of the template can be requested from the OTS and a footstep plan can be executed in case the robot is not able to reach the object with the end-effectors.

3. Once the robot is in a position that the object can be reached by the end-effector the pre-grasp pose can be requested from the OTS and the execution of this step will take the robot's end-effector to a predefined position before approaching the object.

4. Then, the fingers need to be set to any "Open" configuration if available (for safety reasons, in this approach all robots start with fingers in a closed configuration).

5. Afterwards, the grasp pose is requested from the OTS and the execution of this step will take the robot's end-effector to a position ready for grasping the real object.

6. The grasp posture of the fingers can then be set so that the robot takes control of the object (for non-prehensile grasps e.g., turning the valve with a stick, this step can be omitted).

7. If the robot has grasped a floating object, the object template can be attached to the end-effector. It is then considered during collision checking while planning motions and will be moved appropriately.

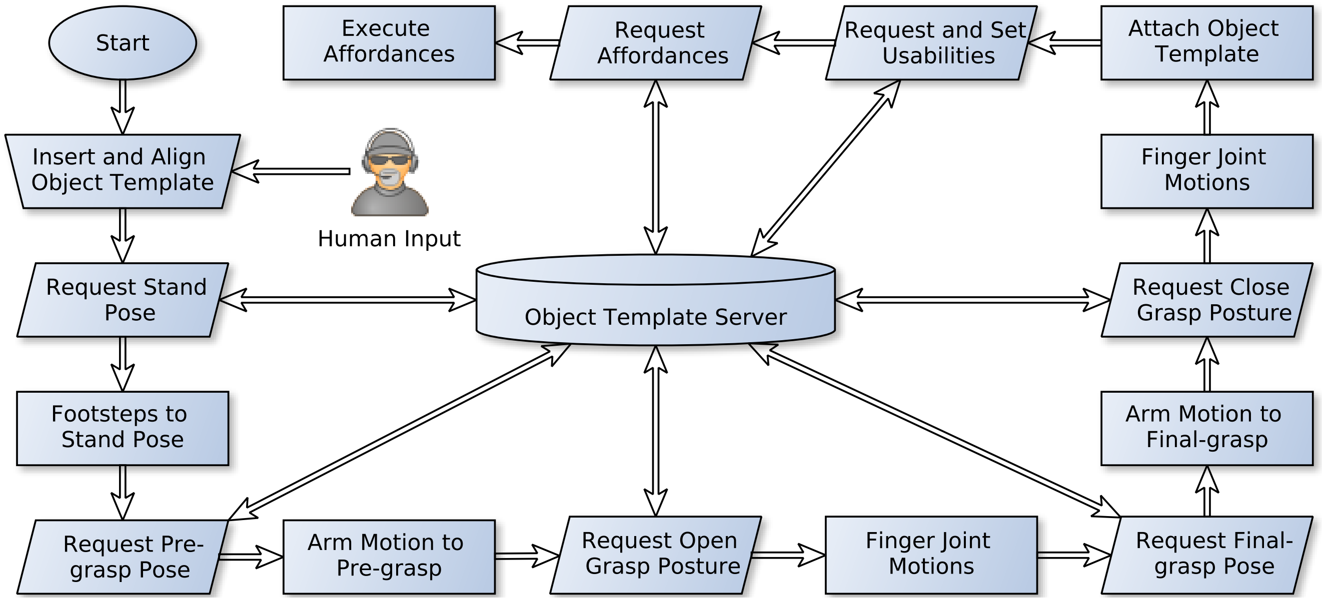


Figure 5.7: Workflow of a manipulation task using object templates. Trapezoids represents manual input that is always perform by a human operator. Parallelograms represent object template data requests to the Object Template Server that can be done either by a human operator or by a high-level behavior. Rectangles represent processes executed by the remote humanoid robot.

8. Then, usabilities of the object template can be selected and used for motion planning assuming this attachment as a rigid body transformation between the origin of the object template and the robot's end-effector.
9. Finally, the affordances of the object template can be requested from the OTS and be executed so that the robot performs the required arm motions to achieve the manipulation task.

After the manipulation task is finished, proper motions to return the object might be required. This can be done by executing the grasp motions in reverse, however, this is not currently considered in this approach.

5.6 A Multi-operator Approach

The task pipeline described in Section 5.5 can be approached with more the one operator. This multi-operator approach allows individual operators to focus in fewer tasks, thus additionally reducing their mental workload. As example, the multi-operator approach used by Team ViGIR during both, the DRC Trials and Finals is described.

The individual operator stations were separate instances of the same user interface that shared data between operators. Thus, if one operator requested a point cloud, the same point cloud would be visible on all stations. This allowed the operators to coordinate verbally with one another, which permitted operation as a "Wizard of Oz" interface where one operator could request of another to gather the additional information needed [45]; this reduced the cognitive load on any one operator. For the DRC Finals, Team ViGIR used four operators with well-defined roles: Supervisory, Primary, Auxiliary, and Immersed operators. Team Hector defined almost the same operator roles except the immersed operator, as Johnny's sensors do not deliver long range 3D data like Atlas. The operators' relative positions in the control room is depicted in Figure 5.8.

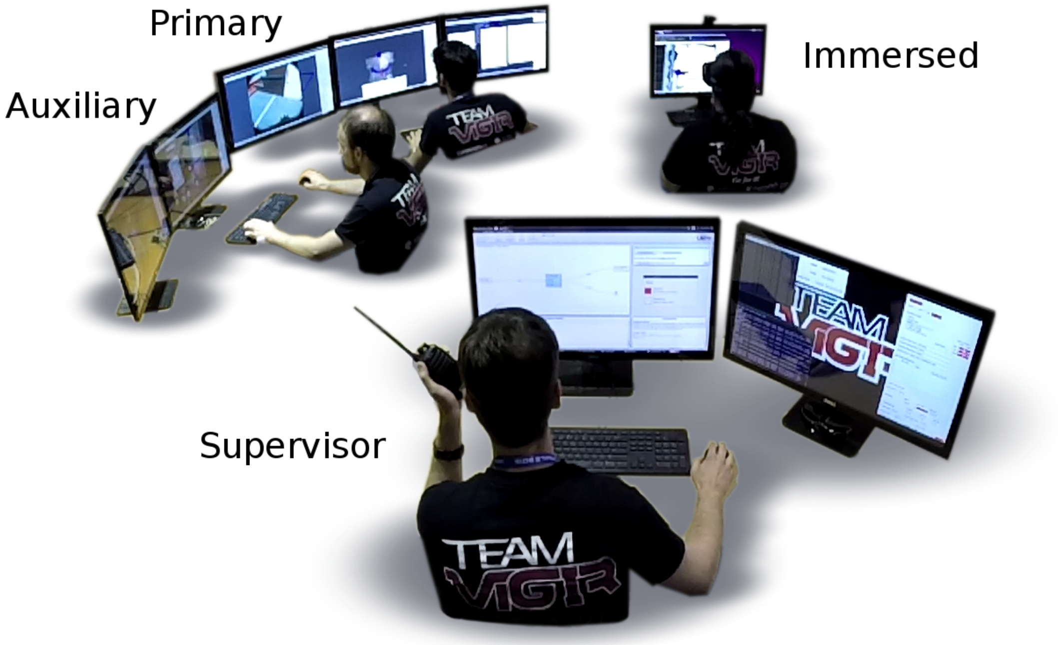


Figure 5.8: The multi-operator approach consists of a supervisor, a primary operator, an auxiliary operator, and an immersed operator.

5.6.1 Supervisor Operator (Planning)

The supervisor operator (or just supervisor) was responsible for coordinating the human-robot team. He was the one that started and stopped the execution of high-level behaviors. He also interacted with the active behavior and adjusted its autonomy via FlexBE's GUI. The supervisor communicated with the other operators verbally, in order to keep them up to date with the progress of behavior execution and to convey requests.

5.6.2 Primary Operator (Action)

The primary operator (or main operator) was responsible for monitoring and verifying actions generated by the active high-level behavior. The primary operator would also intervene and interact with the OCS in order to command the robot to execute actions such as footstep or manipulation planning, if the corresponding behavior state failed. The commands are given using the object template approach, however, if needed, teleoperation is also available as fallback option.

5.6.3 Auxiliary Operator (Perception)

The auxiliary operator was responsible for perception tasks, such as inserting and manipulating the object templates, or other semantic information as requested by high-level behaviors. The auxiliary operator would also gather perception data on the OCS side in support of the primary operator's situational awareness. For Team ViGIR, the auxiliary operator also served as team lead during the runs, and was responsible for making the final decisions on tactics.

5.6.4 Immersed Operator (Observation)

The immersed operator used an Oculus Rift DK220 virtual reality head-mounted display to observe task execution. This permitted him to visually navigate the 3D scene and thus assist in situational awareness. The immersed operator was also able to manipulate object templates, but since visual was dedicated to the head-mounted display, the manipulation of object templates was performed using the 6DOF motion and orientation detection game controller Razer Hydra.

5.7 Collaborative Autonomy

The task pipeline described in Section 5.5 can also be automated using a high-level behavior. Based on the object template concept, this high-level behavior can be used towards an approach called *collaborative autonomy* [79].

Definition: Collaborative Autonomy. *Given a high-level task and a team that comprises a robotic system and any non-zero number of human operators, the team exhibits collaborative autonomy if the robotic system carries out the task autonomously, when capable of doing so, initiates requests to the human operators when necessary, and can respond according to their input at any time. Such autonomy-driven requests include requests for data, requests to perform actions, requests for the operators' permission or confirmation, and decisions that the system wants the operators to make.*

In this context, the robotic system mentioned above is a high-level behavior controlling a humanoid robot in order to carry out a manipulation task. The definition above does not specify the number of human operators, as long as there is at least one. For example, a single operator can take on multiple roles.

This proposed *collaborative autonomy* is related to *collaborative control* as introduced by [28], [27]. Specifically, collaborative control can be seen as a broader definition, which encompasses two concepts of interest: collaborative perception and collaborative autonomy. Briefly, in collaborative perception the human operators assist the robot with perception tasks, e.g. detection of objects of interest, whether in semi-autonomous or fully autonomous operation. In collaborative autonomy, they assist with cognition, decision making, and even actions, such as object manipulation. [42] have identified collaborative autonomy as one of ten challenges for making automation a "team player" in joint human-agent activity. Specifically, in this work, one or more human operators collaborate with a high-level behavior to control a remote humanoid robot.

Collaborative autonomy is also related to the paradigm of supervised autonomy [14]. However, the collaborative autonomy concept presented here also allows for the online adjustment of the level of autonomy, along the lines of [17]. This collaborative autonomy allows a human-robot team to carry out a task together; with the high-level behavior autonomously requesting operator input when required and the operator intervening, if deemed necessary, and then handing control authority back to the behavior.

In this approach, the primary and auxiliary operators act as part of the autonomous behavior. The supervisor commands the execution of behavior, and in case of failure in a state, the primary operator can act as a substitution for that particular state. Then, the primary operator will execute the corresponding actions of that state and will return control to the autonomous behavior when finished. This collaborative autonomy design led to the observation that the human operator can be seen as just another system capability from the point of view of a high-level behavior. That is, a high-level behavior can be somewhat agnostic to whether an action was performed automatically by the robot or manually by the operator, as long as the necessary postconditions are met.

The design philosophy behind this high-level behavior control approach is based on the challenges that a humanoid robot would face while carrying out remote search and rescue operations in collaboration with human operators. On the one hand, communication and time constraints mean that one could not rely solely on the human operators to make every decision and perform every action. On the other hand, leveraging the human operators' cognitive and perceptual capabilities is also desirable. This gave rise to the concept of *collaborative autonomy* described above. Furthermore, the degraded communications between operators and the robot motivate interaction between them at a higher level of abstraction. A significant level of abstraction via the use of object templates and affordances is achieved because this information can be used by human operators, the robot, and the high-level behavior. This collaborative autonomy approach focuses in abstracting the interaction between the operators and the robot at the task-level. The high-level control philosophy described above has been implemented in FlexBE [96] the Flexible Behavior Engine [94, 95].

In Section 5.5 it was discussed how the operator can use teleoperation and object templates in order to command robot motion execution. Here it is now shown that high-level behaviors can also leverage the manipulation planning subsystem. This introduces another layer of abstraction and enables task-level autonomy. In brief, this system design allows carrying out the object-template-based manipulation workflow depicted in Figure 5.7 programmatically. The steps of the workflow are implemented as finite state automata and the workflow itself is realized as part of a hierarchical finite state machine. The state implementations do not carry out manipulation-related computations themselves. Rather, they interface with manipulation system subcomponents, such as the OTS (on-board) and the manipulation planning backend.

The level of autonomy of the high-level behavior can be set, for example, at four different levels: `Off`, `Low`, `High`, and `Full`. These levels reciprocally correspond to the amount of intervention that a human supervisor will have during task execution. An `Off` level of autonomy will mean that there is no autonomy at all, thus, the operator is required to take control over every action, in other words, using the object template approach in a manual way. A `Low` level of autonomy will mean that the high-level behavior is expected to execute all the tasks, however, every single transition will require the confirmation of the supervisor. A `High` level of autonomy will mean that the high-level behavior is expected to execute all the tasks, but in this case, some transitions will require the confirmation of the supervisor and other transitions will be automatically done by the high-level behavior. Finally, a `Full` autonomy level will mean that the high-level behavior will not request any confirmation of the supervisor and will attempt to perform the complete task autonomously. The `Full` autonomy level currently still requires that the first step in the workflow, i.e., the object template insertion and alignment, gets performed by the human supervisor. Since the cognitive abilities of the supervisor for object detection are leveraged, the behavior can requests the operator to identify the object of interest and then insert and align the appropriate template. It would be possible to incorporate autonomous algorithms for template alignment into collaborative autonomy, but this still an open topic for research at this time. After the template has been aligned, the high-level behavior can autonomously continue execution of the rest of the steps depending on their level of autonomy. Supervisors can then provide confirmation, assist, or intervene whenever the case arises.

6 Experiments, Applications, and Results

This chapter presents a series of applications where the contributions presented in this thesis have been applied as well as laboratory experimentations designed to demonstrate the particular aspects of the approach presented in this thesis showing how it contributes to the state of the art. Also, this chapter presents an analysis of the performance of two different robotic systems during participation of the renowned international competition “DARPA Robotics Challenge” which during the last three years has been considered the most ambitious and challenging robotic competition.

Laboratory experiments were designed and performed to show the capabilities of the particular aspects that differentiate this contribution from the current state-of-the-art approaches. Specific experimentation using object template affordances and object template usabilities is presented. A demonstration of execution of manipulation tasks outside the workspace of supervised robots using objects as online-augmented end-effectors is also shown.

6.1 Robot Hardware Used for Experimentation

In this section the two robotic systems as well as the end-effectors used during this project are presented. Even though there exists several types of rescue robots as described in Chapter 2, this approach focus on using the advantages of humanoid robots over tracked and wheeled systems. In case of disaster in human-designed environments, most of the elements found are engineered for humans (even when degraded). Door handles and light switches are located near the hand height, steps in the stairs are designed to be comfortable for the average human leg length and the width of doors, stairs, and floors is also designed according to human size. Finding human tools is highly probable and considering a human operator, the ability of planning with respect to a humanoid form is more natural for domain human experts. In this approach these characteristics are found to be relevant for the design of robots and end-effectors which rescue or exploration missions can take place in human environments.

6.1.1 Atlas

Atlas is a humanoid robot and has been developed by Boston Dynamics Incorporated (BDI) in the USA. This robot has been upgraded in several ways starting on August 2013 and the last version described in this thesis on July 2015. For simplicity only two of these versions, version 3.0 (v3) used in 2013 and version 5.0 (v5) used in 2015, are considered.

The Atlas robot v3 was completely actuated using hydraulic servomotors. It had 28 DOF, 6 per leg, 6 per arm, 3 in the torso, and a “tilt” joint in the head. This version weighed 150kg and was a tethered robot which means that power and communications were provided using a cable. External sensing was performed using a Multisense SL sensor head, which provided both a stereo camera and a continuously spinning LIDAR. Given that Atlas had no “pan” joint on the neck, two additional fisheye cameras were located near the chin of the robot. These wide-angle cameras were pointing sideways for providing situation awareness to the operator(s). Internal sensing is performed using a high performance KVH 1750 IMU using Fiber Optic Gyroscope (FOG) technology. The data from this IMU system was used to estimate robot odometry with high accuracy; additionally, it was fused with leg kinematics to increase this estimation. Additionally, the robot was equipped with 6 DOF force/torque sensors in the wrist and 3 DOF force/torque sensors in the ankles. An Atlas robot v3 was given to each of the top six teams that competed in the Virtual Robotics Challenge (VRC) which at that time was considered one of the most advanced anthropomorphic robotic systems in the world.

The Atlas robot v5 was upgraded given the feedback gathered and lessons learned during operation of Atlas v3. With only 6DOF arms, Atlas v3 suffered of poor manipulability in the visible workspace of the robot. For this reason, 3 electric joints per arm were added, making each arm a 7DOF manipulator. Also, the hydraulic joints in the thighs were replaced and received a new hydraulic pump with variable pressure. After all the upgrades, Atlas v5 weighed 180kg. This was mainly due to the unplugged upgrade which consisted of adding a battery backpack to allow for at least 1 hour of electric autonomy.

Operational use of the Atlas robot in all its versions required high safety procedures. For instance, working with high-pressure oil presents potential burning danger. Additionally, the torque generated by the hydraulic joints in combination with the high mass of each extremity of the robot can produce high inertias that can be harmful for humans bear the robot. A $3m$ safety radius around the robot needed to be clear from humans when the hydraulic pump was running.

Figure 6.1: Two of the humanoid robots used: the Boston Dynamics Atlas robot (left) and the Robotis THORMANG robot (right).

6.1.2 THORMANG

THORMANG is a humanoid robot system which has been developed by Robotis in South Korea [120]. This robot has been upgraded in several since 2013, the version referred in this thesis is version 1.0 (v1). This robot is completely actuated using the electric servomotors "Dynamixel Pro". It has 30 DOF, 6 per leg, 7 per arm, 2 in the torso, and 2 in the head. THORMANG weighs around 50 kg and stands 1.5m tall. External sensing is performed using an oscillating Hokuyo UTM-30LX EW LIDAR sensor in the chest and a RGB camera on the head. Internal sensing is performed with a Microstrain IMU used as an Attitude and Heading Reference System (AHRS). Additionally, the robot is equipped with 6 DOF force/torque sensors in the wrist and ankles. THORMANG has an electric autonomy of more than 1 hour.

Table 6.1 shows a comparison of the two humanoid robots used for experimentation.

Robot Type	*Atlas v5*	*THORMANG v1*
Mass	180kg	50kg
Height	1.9m	1.5m
Number of DOF	30	30
Actuation	Hydraulic and Electric	Electric
Internal Sensing	KVH 1750 IMU	Microstrain IMU
External Sensing	Hokuyo UTM-30LX LIDAR FPGA stereo system 2 Fisheye cameras	Hokuyo UTM-30LX LIDAR Monocular camera
Force Sensing	6DOF wrist 3DOF ankle	6DOF wrist 6DOF ankle
Onboard Computation	3 Intel Core i7 PCs QNX RTOS based control PC	2 AMD 1.6GHz
Connectivity	10 Gbps fiber optic Ethernet WiFi 802.11n	1 Gbps Ethernet WiFi 802.11n

Table 6.1: Comparison table of Atlas v5 and THORMANG robots.

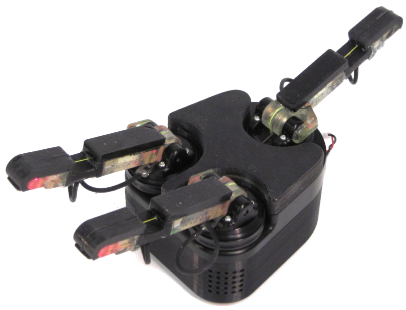
6.1.3 Different End-effectors and Hands

For the Atlas robot, 5 different end-effectors were investigated, however, just four of them were used during experimentation. The Sandia Hand (Figure 6.2a) has three fingers and a thumb, each of them consisting of 3 actuated DOF and two cameras in the palm [91]. The iRobot hand (Figure 6.2b)has two fingers consisting of 2 actuated DOF and one under-actuated joint, and one thumb consisting of 1 actuated DOF and 1 under-actuated joint [68]. The Robotiq hand (Figure 6.2c) has two fingers and a thumb [74]. The two fingers consist of 1 actuated DOF, one shared actuated DOF for spread, and 2 underactuated joints. The thumb consist of 1 actuated DOF and 1 under-actuated joint. The BDI Hook (Figure 6.2d)consisted of a bended metal bar attached to the last link of the arm. This hook proved to be robust to manipulation tasks that were subject to high forces, e.g., supporting the weight of the robot when holding the stairs rail for climbing. Finally, a poke stick attachment (inspired by the hook) (Figure 6.2e). This poke stick allowed simultaneous use with one of the robotic hands previously mentioned.

The iRobot hand and the hook were used with Atlas v3 (the iRobot hand was used for holding objects and the hook was used for turning the valves). The poke stick was simultaneously used with the Robotiq hand for the Atlas v5 (the Robotiq hand was used for holding objects and the poking stick was used for turning the valve and potentially activating the ON switch on electric tools)

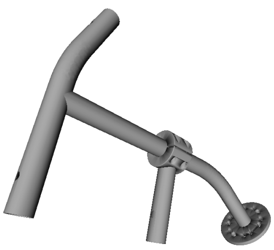
For THORMANG, the hands used were designed by Michael Rouleau [87],[86]. This THOR hands (Figure 6.2f) have two fingers, each of them consisting of 1 actuated DOF and 1 under-actuated joint.

(a) Sandia.



(b) iRobot.

(c) Robotiq.



(d) BDI Hook.

(e) Poking Stick.

(f) THOR.

Figure 6.2: Different end-effectors and hands.

The object template approach is capable of working with these different kinds of end-effectors and robotic hands. Before run-time, the robot will load a description file indicating which end-effector had been attached to the arm and the system automatically loaded the corresponding grasp library for that particular end-effector. This allows the system to be highly flexible if the need for changing the end-effector arises. Table 6.2 shows a comparison of the investigated end-effectors and hands.

6.2 DARPA Robotics Challenge

Executing robotic manipulation tasks in remote and potentially degraded environments presents challenging problems as shown in 2011 during robot operations of the post-disaster Fukushima Daiichi Nuclear Plant in Japan [65]. This showed the lack of robotic technology to help in these situations and motivated DARPA to create the DRC [18] to push technology so that remote robots are capable of performing mobility and manipulation tasks in human-designed but

End-effector	*Total Fingers*	*Total DOF*	*Actuated DOF*	*Under-actuated DOF*
Sandia	4	12	12	0
iRobot	3	8	5	3
Robotiq	3	10	4	6
BDI Hook	0	0	0	0
Poke Stick	0	0	0	0
THOR Hands	2	4	2	2

Table 6.2: Different end-effectors and hands.

unstructured and partially-degraded environments. The DRC was created with focus on the development of rescue robots that can perform tasks to help in the mitigation of disaster scenarios that might be hazardous for human beings to explore.

The author participated in the DRC as part of Team ViGIR[109] with the highly advanced humanoid robot Atlas shown in Figure 6.1. Team ViGIR is a cooperation between research groups in Germany and other research institutions in the USA. Participation in the three main events, the VRC [44] in June 2013, the DRC Trials in December 2013 [45], and the very recent DRC Finals in June 2015, demonstrated that the proposed object template manipulation approach can be used to perform manipulation tasks in unstructured environments using human supervision of a remote semi-autonomous robot.

The three main DRC events — Virtual Robotics Challenge in June 2013, DRC Trials in December 2013, and DRC Finals in June 2015 — focused in simulating disaster scenarios that required robots to perform mobility and manipulation tasks under severe communication constraints.

6.2.1 VRC

The VRC was an online competition in July 2013 between research groups from all around the world. It was designed to be a pure software challenge using a simulated version of the Atlas in three different simulated disaster environments. This simulated environments were developed in Gazebo [69] simulator using the DRC simulator DRCsim. The VRC included mobility and manipulation challenges such as traversing rough terrain, driving a vehicle, and manipulating a fire hose. Although traversing the rough terrain and driving a vehicle were challenging tasks, they did not consider fine manipulation. For this reason, this section focuses on the approach taken during the manipulation challenge of attaching a fire hose to a water pipe in the VRC. A complete overview of the overall approach taken for the three tasks is available in [44].

The manipulation task consisted of approaching to a table located away from the robot, picking up a fire hose placed over the table, attach it to a water pipe located near the table, and finally turning a valve that was located at a short distance from the table. A view from the simulator can be seen in Figure 6.3 and [80] shows an example of lifting the fire hose from the table. Scoring was divided in four steps, each giving a point: raising the fire hose from the table, aligning it to the pipe, attaching it by turning the fire hose clockwise, and the final point was giving by turning the valve. This task was performed in 5 different instances and it was required to be completed in 30 minutes of simulation time.

For the VRC the object template manipulation approach presented in this thesis was still in an earlier state, there was no affordance information available to execute manipulation at a task level. For this reason, only grasp positions for the end-effectors were considered in the VRC; manipulation was performed using the grasp in the object template and changing its pose to generate target poses for the end-effector.

Constrained communications consisted of a simulated uplink and downlink to the Gazebo remote server where simulation was running. Limitations were defined as a budget of transferred data through the links and also latency was added.

The top six teams resulting from this challenge would be given by DARPA a real Atlas robot for the next challenges. From the 126 teams registered for the VRC, only 26 qualified to participate and only 22 scored points. Team ViGIR was ranked $6th$ with 27 points and in September 2013 was given a Government Furnished Equipment (GFE) Atlas to participate in the DRC Trials. Table 6.3 shows the results of Team ViGIR in the manipulation task and Table 6.4 shows the points in the manipulation task from the top six teams.

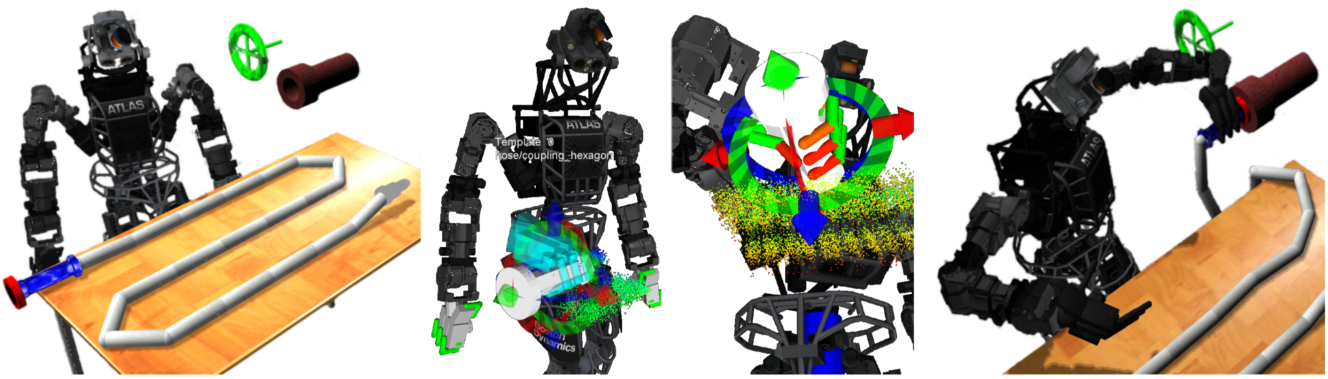


Figure 6.3: VRC manipulation task view from Gazebo simulator and the OCS. From left to right: Gazebo Hose setup with water pipe and valve. The Hose Template aligned with the pointcloud data of the real hose showing the blue ghost hand in the OCS. The robot grasping the hose with finger tactile information in the OCS. Gazebo view of the robot aligning the hose to the water pipe.

	Run 1	*Run 2*	*Run 3*	*Run 4*	*Run 5*	*Total (Max. 20)*
Manipulation Task	1	1	2	1	1	6

Table 6.3: Team ViGIR VRC scores for the manipulation task.

6.2.2 DRC Trials

The DRC Trials 2013 were held in Homestead, FL, USA. Research groups from different countries participated in a series of tasks to demonstrate robot capabilities for rescue missions. These tasks considered robot capabilities such as mobility and manipulation in disaster environments like:

- Walk through rough terrain,
- Climb up a ladder,

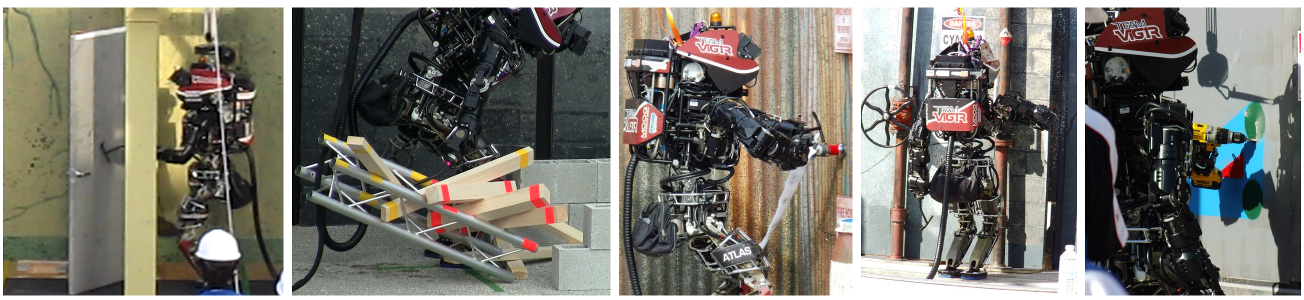
Place	*Team*	*Manipulation Task*	*Total (all tasks)*
1^{st}	IHMC	20	52
2^{nd}	WPI	4	39
3^{rd}	MIT	9	34
4^{th}	TRACLabs	6	30
5^{th}	JPL	4	29
6^{th}	ViGIR	6	27

Table 6.4: Top six teams VRC results.

- Remove debris blocking a doorway,
- Open three different types of doors,
- Break through a wall using a cutting tool,
- Attach a fire-hose to a wye,
- Close three different types of valves and
- Drive a car.

Each of these tasks were to be performed by real robots in thirty minutes each. Figure 6.4 shows examples of the manipulation tasks performed by Team ViGIR. The driving task was determined to require significant development effort that would not be re-usable for much of the other tasks, for this reason, this task was not attempted. The walking through rough terrain task and climbing up the ladder were tasks that did not focus in manipulation, for this reason they are omitted. A comprehensive description of the performance in all the DRC Trials tasks is available in [45].

While a single operator was used for the VRC, two operators interacted directly with the robot, with the auxiliary operator being responsible for managing perception data and the primary operator responsible for commanding robot motion via teleoperation or task-level commands.



(a) (b) (c) (d) (e)

Figure 6.4: Manipulation tasks in the DRC Trials: (a) Opening the first door, (b) Pulling the truss out in the debris task, (c) Attempting to connect the hose to the wye, (d) Rotating the first valve, (e) Using the drill to break through the wall.

Each of the tasks consisted of three defined checkpoints. A point was given for each accomplished checkpoint and in case all checkpoints were accomplished without an intervention (robot failure which required a restart) a bonus point was given. The tasks required different amounts of mobility and manipulation and for the purposes and scope of this thesis only the Hose task and the Valve task are described because they contain good examples where the object template manipulation approach has been applied. A comprehensive description of the results obtained by Team ViGIR for these tasks has been published in [45].

Hose Task

In the Hose task, the robot needed to walk to a reel and pick up a fire hose, then walk with the fire hose towards a wye and attach it by turning the nozzle (Figure 6.5a). The first point in this task was obtained when the robot crossed the yellow line on the floor while carrying the hose. The second point was given when the hose came in physical contact with the wye and the third point was obtained for attaching the hose. The approach taken to accomplish this task was first to divide it in three subtasks: pick up the hose, touch the wye with the hose and attach it. Figures shown in this section contain screenshots from the OCS, either a top-down view of the environment or a 3D view. The figures also contain images of the real scenario, obtained from different cameras located in the walls of the task (Figure 6.5a).

Pick up the hose

Following the pipeline described in Section 5.5 the task started by acquiring sensor data information from the environment. The secondary operator requests a pointcloud of the reel and inserts a hose template aligning it to the 3D data belonging to the real hose (Figure 6.5b). Then the primary operator requests a footstep plan to a position relative to the hose template where the robot can easily grasp the hose as shown in Figure 6.5c and Figure 6.5d. Once the robot is standing in front of the hose, the primary operator requests a grasp to pre-grasp pose of the ghost hand (Figure 6.6a and Figure 6.6b), then the robot moves the hand to the final-pose and executes the grasp (Figure 6.6c and Figure 6.6d).

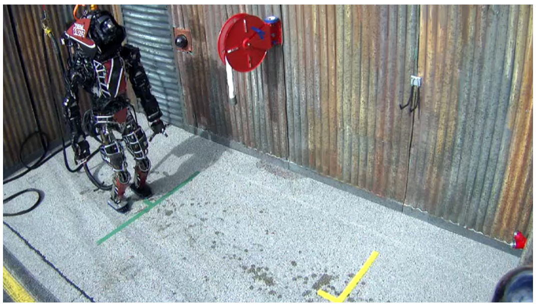
After grasping the hose the primary operator commands the robot to move one meter to the right based on an environment map previously requested by the secondary operator (Figure 6.7).

Touch the wye with the hose

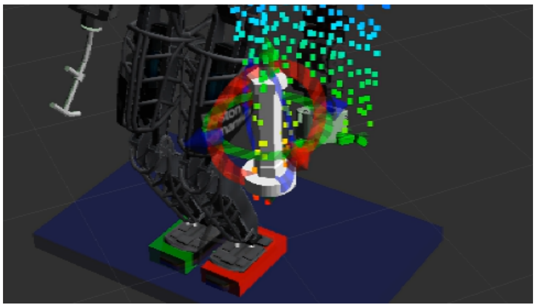
For this subtask the same pipeline was applied again. The secondary operator requests a pointcloud of the wye (Figure 6.8a) and inserts the wye template, aligning it to the 3D data belonging to the real wye (Figure 6.8b). Then the primary operator requests a footstep plan to a position relative to the wye template where the robot can easily touch the wye with the hose (Figure 6.8c and Figure 6.8d). Once the robot is standing in front of the wye, the primary operator request the robot to move the hand to the pre-grasp pose of the ghost hand (Figure 6.9a), which the robot executes (Figure 6.9b and Figure 6.9c). Then the robot moves the arm to the final-grasp pose which makes the hose to come in physical contact with the wye (Figure 6.9d).

Attach the hose

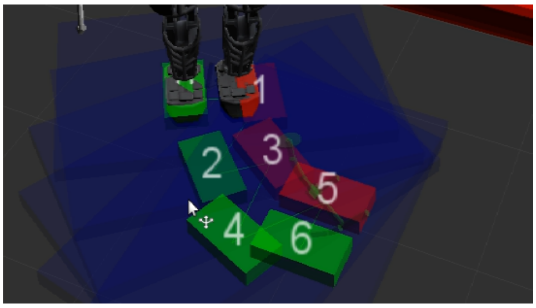
At this point the object template manipulation approach had been successfully applied to manipulate and align the fire hose to the wye and the only missing thing was turning the nozzle



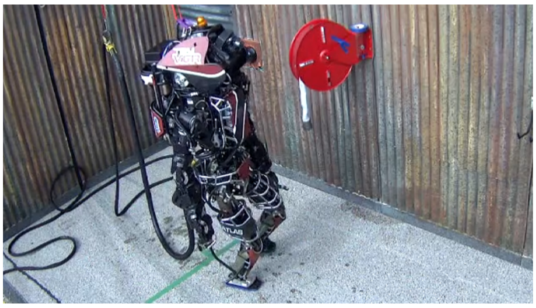
(a) Hose setup.



(b) Hose poincloud and template.

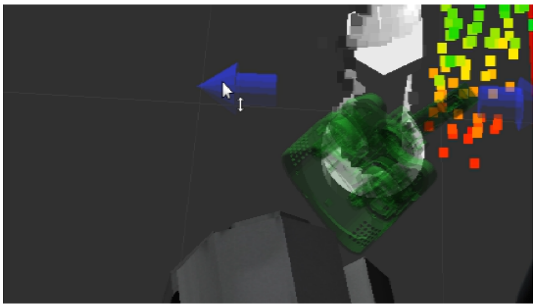


(c) Footstep plan to Hose.

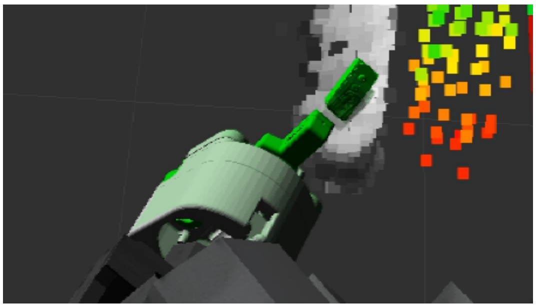


(d) Robot executing plan.

Figure 6.5: Request to walk to the hose. (a) Robot start position, hose reel and wye. (b) Reel and hose pointcloud with the template of the hose. (c) Footstep plan visualization to the hose in the OCS before walking. (d) Robot following the footstep plan.

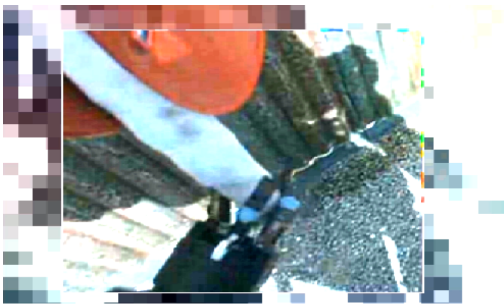


(a) Hose ghost hand.



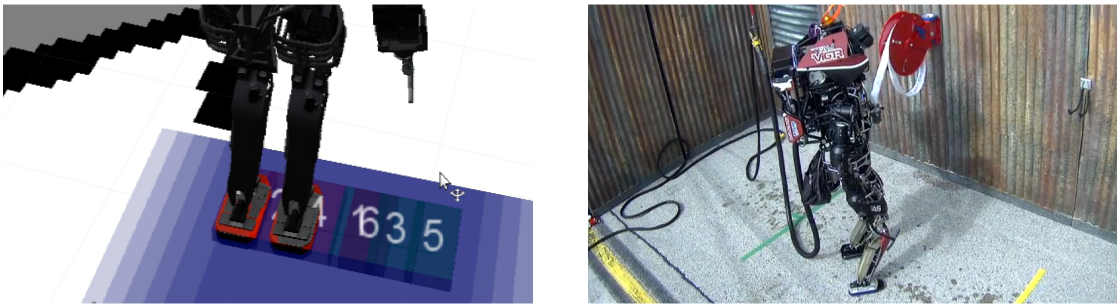
(b) Robot executing plan (OCS).

(c) Robot executing plan.



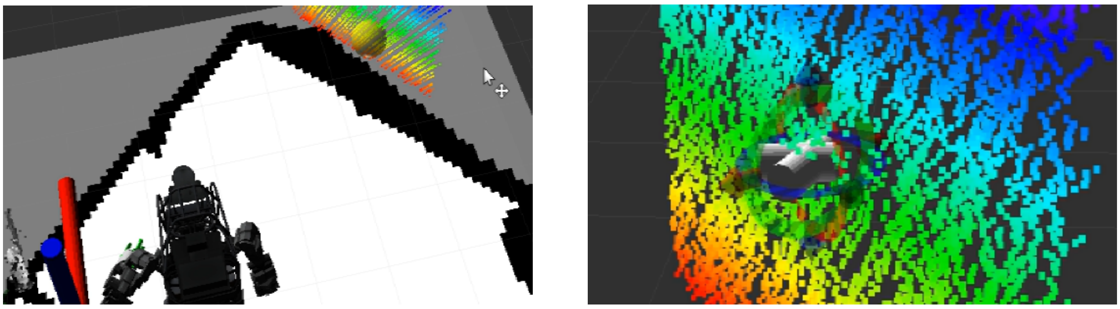
(d) Robot grasping hose.

Figure 6.6: Request to grasp hose. (a) Hose ghost hand visualization for the operator to verify. (b) and (c) Automatic motion of the robot to place the hand in final-pose. (d) Operator verifies the grasp through robot's camera.

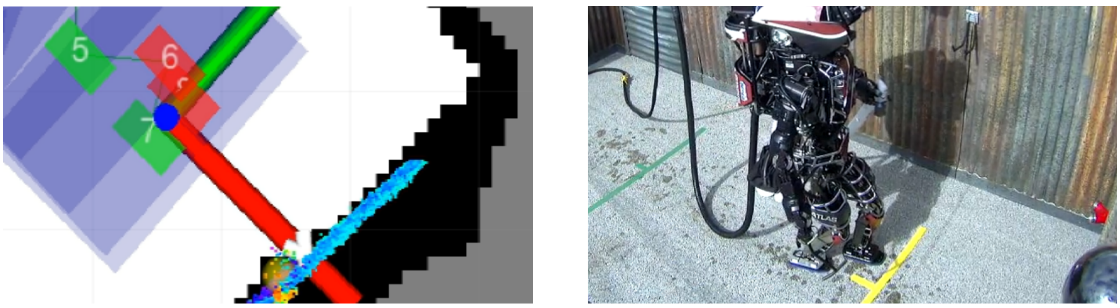


(a) Footstep plan away from reel. **(b)** Robot executing plan.

Figure 6.7: Request to walk away from reel. (a) Footstep plan with lateral right steps. (b) Robot walking with the hose.

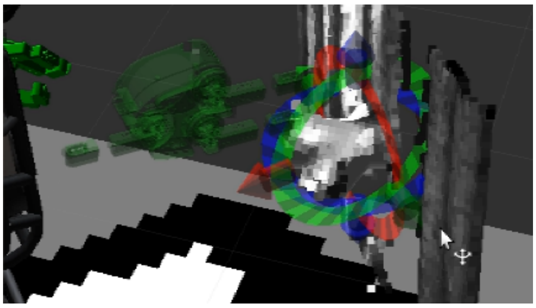


(a) Wye pointclod. **(b)** Wye template aligned.

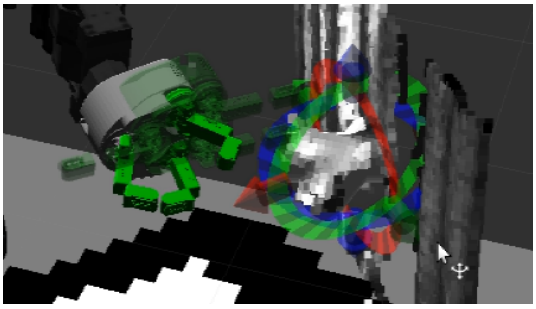


(c) Footstep plan to wye. **(d)** Robot executing plan.

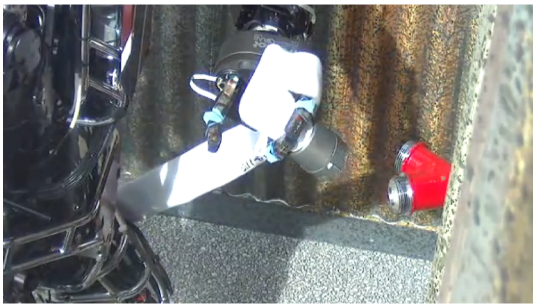
Figure 6.8: Request to walk to the wye.



(a) Wye ghost hand.



(b) Robot moving arm to ghost hand.



(c) Robot executing plan.

(d) Hose and wye make physical contact.

Figure 6.9: Robot commanded to bring the hand near the wye and then touch it.

to engage the hose. Given the extremely small size of the nozzle bumps used to turn it (around 0.25 cm^3), this subtask was not feasible to solve using this approach. Teleoperation was used instead, however, this was a fine manipulation task which required high precision and even the hose was correctly located (Figure 6.10a) and the nozzle was turned 180 degrees (Figure 6.10b), the threads of the wye and the hose did not engage, and the hose fell after releasing it.

(a) Hose aligned.

(b) Robot turning the nozzle.

Figure 6.10: Attach hose and turn nozzle (teleoperated).

Hose Task Results

To compare the results obtained using this approach, the performance of other teams using the Atlas robot during the DRC Trials has been analysed. Table 6.5 shows the timetables of the activities performed during the Hose Task day one [20] and day two [21].

Activity	***Team ViGIR***	*IHMC*	*MIT*	*TRACLabs*	*WRECS*	*Trooper*	*HKU*
Task Start Time (min)	00:00	00:00	00:00	00:00	00:00	00:00	00:00
Stand in front of Hose	00:40	01:15	02:15	01:00	01:00	03:25	03:40
Picked Hose	04:00	08:40	03:53	06:20	03:10	14:40	-
Crossed Line (point)	**08:00**	11:50	09:30	08:45	07:55	29:30	-
Stand in front of Wye	08:30	13:25	11:00	09:15	08:18	-	-
Touched Wye (point)	**10:10**	15:20	15:25	16:25	11:28	-	-
Hose alignment	18:00	16:20	22:35	29:00	-	-	-
Turned Nozzle	**22:20**	23:00	-	-	-	-	-
Points	2	2	2	2	2	1	0
Overall Rank	9^{th}	2^{nd}	4^{th}	6^{th}	6^{th}	8^{th}	10^{th}

Table 6.5: Hose Task timetable for all Atlas teams.

These results show that Team ViGIR was the fastest to perform manipulation tasks. Team ViGIR was able to command the robot to walk with the fire hose through the yellow line in the floor within 8 minutes, touch the wye at time 10:10, align it with the wye at 18:00 and start turning it at time 22:20. From these results it can be seen that there were only two teams able to turn the nozzle of the hose, and in most of the other manipulation activities Team ViGIR was faster than other teams.

6.2.3 DRC Finals

The DRC Finals was the last of these events taking place at Pomona, California on June 5th and 6th 2015. In contrast to the VRC and the DRC Trials, all tasks had to be performed in a single run. In addition, the robots had to be untethered. Each participating team's robot had sixty minutes to perform the eight tasks — driving a vehicle, egressing the vehicle, opening a door, turning a valve, breaking a wall using a tool, performing one of three surprise manipulation tasks, traversing rubble or uneven terrain, and climbing a ladder. Each team was allowed two runs in the competition, one on the first day and one on the second competition day. The DRC Finals presented challenges in a wide range of areas. On one side, the robot system capabilities to complete the individual tasks designed for the challenge were put to the test. On the other side, the challenges of real-world field scenarios such as temperature, ground slope, and communication interferences among others, pushed the humanoid robots to their limits.

Robots were required to reach the door either by walking or driving a distance around $60m$. Before attempting any other task, robots needed to open the door. Including the door, 4 tasks required manipulation skills. To keep the focus on the experimental evaluation that concerns this thesis, the first task of the DRC Finals which corresponded to driving a vehicle is omitted.

Scoring in the DRC Finals was awarded 1 point for achieving each tasks. However, walking instead of driving prevented the teams from getting points for driving and egressing the car. Additionally, teams that chose to drive were allowed to skip the egress task by requesting a reset (meaning starting since the last point in a walking position plus a 10 min penalty).

Communications were only degraded after traversing the door, simulating an indoor environment. Gradually, communications improved and degradation stopped 15 minutes before the run end.

A total of three teams that participated in the DRC Finals were using software that included the object template concept implementation presented in this thesis (Team Hector in Section 6.2.3, Team ViGIR in Section 6.2.3, and Team Valor in Section 6.4). Two of these teams were able to open the door using the object template concept, and one of them successfully completed the valve task. Thus demonstrating the capabilities of the contributions of presenting in this thesis. However, due to hardware issues in the robots from these three teams, use of all the aspects of the object template concept was not able to be shown during the DRC Finals. For this reason, laboratory experimentation was performed after the DRC Finals in order to demonstrate the capabilities of the unique aspects of the concept presented in this thesis. This will be presented in Section 6.3.

Team Hector

Opening the Door: Day 1

During day one, Team Hector also opened the door using a combination of high-level behaviors, object template manipulation, and teleoperation.

The first attempt was mainly commanded by the high-level behavior. The auxiliary operator placed the door template based on captured point cloud data. Using the template pose information, the high-level behavior was able to generate the transitions to move the left hand to the pre-grasp and afterwards to the grasp pose. The primary operator noticed a significant offset through the camera images which was likely caused by sensor noise and non-perfect LIDAR to robot frame calibration. Several iterations between the auxiliary operator to perform template adjustment and the supervisor operator to generate manual transitions in the high-level behavior were required to bring the robot's hand into the grasp pose of the door handle.

Due to a hardware failure, Team Hector was not able to continue execution of the arm motions to open the door, this forced them to request a reset. After this reset, the primary operator intervened and continued the task executing Cartesian teleoperation to re-grasp the door handle. Once the robot's hand was confirmed to have grasped the door handle, the primary operator requested the execution of the "Turn Clockwise" affordance of the Door Template. In this way, Team Hector managed to open the door.

The concept of work sharing between operators and robot worked well, as all operators had been able to provide all needed information to the behavior. This additionally shows that the high-level software architecture developed initially for the Atlas robot was successfully implemented on the THORMANG humanoid robot. The lack of recorded information from the task, prevents the author from providing high-detail information; in lieu of an explicit timeline, Figure 6.11 presents a series of images from the robot executing the door task.

Team ViGIR

The performance for the three manipulation tasks attempted during the two competition days is briefly described the next paragraphs.

Opening the Door: Day 1

During day one, Team ViGIR opened the door using a combination of high-level behaviors, object template manipulation, and teleoperation. The auxiliary operator identified the door and

Figure 6.11: DRC external footage from the door task. From left to right: Setup, Pre-grasp, Grasp, Open affordance, Door open.

inserted the door template, aligning it so that the door-handle matched. The high-level behavior requested to the auxiliary operator the ID of the template to start the task. First, the high-level behavior requested from the object template server the stand pose to open the door, which was used as input to request a footstep plan for stepping towards the door. The high-level behavior commanded the robot to walk to the stand pose; the robot responded to these commands, but the response took longer than expected.

After reaching the stand pose and autonomously positioning the torso to face towards the door, the behavior's attempts at requesting an arm motion plan to the door handle's pre-grasp pose failed repeatedly. It was later discovered that this was due to the unplanned[1] degradation of communications between the field computer (where FlexBE, i.e., the active behavior, was running) and the computers on-board Atlas (where the planning system was running). Regardless, the supervisor communicated the failures to all operators and asked the primary operator to take over.

The primary operator commanded the robot to move the arm to the pre-grasp pose. Misalignments from the door template prevented the robot to reach the grasp pose. Because of this, the primary operator proceeded to execute the arm motion using Cartesian teleoperation. Once the robot's hand grasped the door-handle, the primary operator requested the "Turn Clockwise" affordance of the door template. However, the turning motion of the door-handle was not enough to unlatch the door. So when the primary operator requested the "Push" affordance, the grasp from the door-handle was lost. The primary operator proceeded to perform Cartesian teleoperation and the door was opened by pushing downwards on the door-handle. Figure 6.12 shows the series of images from the robot's camera view of the execution of the task. Figure 6.17a shows a timeline of the events required to perform the task.

As can be seen, initially, the supervisor operator was able to command the robot using high-level behaviors. However, after a failure, the primary operator intervened for the rest of the task using object template control (grasp and affordance commands) and Cartesian teleoperation. This experiment shows the cascade of changes in the control approach taken by the operators to achieve the task. Given the flexibility to change approaches on the fly, it is shown how the operators were able to adapt to higher-layer system failures and use lower layers of manipulation control to open the door.

After the 10 minute penalty time, the door task was attempted. During the attempt to perform the door task, the supervisor team noticed that high level behavior execution did not work as

[1] DARPA's planned degradation of communications was between the operator control station and the field computer(s).

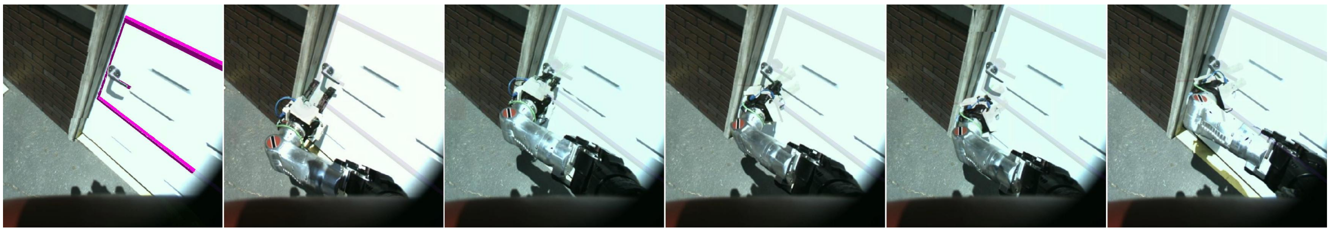


Figure 6.12: Robot's camera view from the door task. From left to right: Door template aligned, Pre-grasp, Grasp, Turn Clockwise affordance, Push affordance (fails to open), door opened after Cartesian teleoperation.

intended. This was later traced back to a faulty setup of the communications bridge system and increased saturation of the wireless links used in the competition. The supervisor team thus switched from using assisted autonomy via Flexible Behavior Engine (FlexBE) behaviors and use of object templates to using object templates and teleoperation. Using this approach, the door was successfully opened as visible in (Figure 6.12). The valve task was solved using mainly object affordance level control (Figure 6.13). Before being able to actuate the switch in the surprise task, time ran out, ending the run. A video is available online [19].

Turning the Valve: Day 1

After traversing the opened door, the next task was to open a valve 360 degrees counter-clockwise. At this point, there was no possibility to use high-level behaviors due to communications issues; for this reason the primary and auxiliary operators performed the rest of the tasks.

The auxiliary operator inserted and aligned the template to match the pose of the real valve. The primary operator requested the stand pose from the template and the footstep plan was calculated. It was discovered that given the communication constraints, footstep plans with more than 10 steps were not able to be visualized in the OCS. For this reason, manual creation of footstep plans was required to reach the valve. Once the robot stood in front of the valve, the primary operator successfully used grasp commands to place the robot's wrist attachment (a poking-stick located in the wrist of the left hand) in-between the crossbars of the valve. Afterwards, the operator used the "Open" affordance of the valve template to turn the valve. Figure 6.13 shows the series of images from the robot's camera view of the execution of the task. Figure 6.17b shows a timeline of the events required to perform the task.

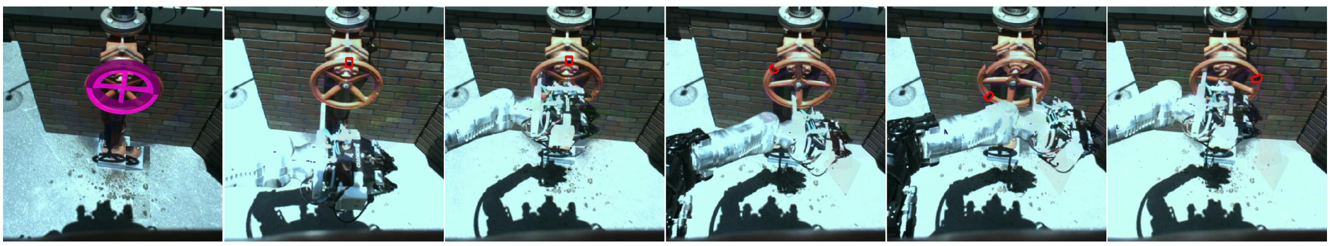


Figure 6.13: Robot's camera view from valve task. From left to right: Valve template aligned, Pre-grasp, Grasp, Open affordance 45, 135, and 270 degrees.

Opening the Door: Day 2

On day two, the behavior started by requesting that the door template be placed and aligned. The auxiliary operator responded by providing this information. The behavior was then able to request and autonomously execute a footstep plan towards the door template's stand pose. Since the communications issue from day one (by moving FlexBE to the on-board computers) were addressed, the supervisor was confident enough to switch the behavior's current autonomy level to `High`. The behavior had the robot look down towards the door handle and turn its torso. It then requested the auxiliary operator to adjust the door template, if necessary. Afterwards, the behavior autonomously moved the robot's arm to the pre-grasp pose and then the grasp pose. Once the hand moved to the grasp pose of the door template, the behavior asked for permission to proceed. The primary operator noticed that the robot's hand was not in the correct pose. After adjusting the hand (using the affordances of the door template), the supervisor gave the high-level behavior permission to proceed. Had it not been for this application of collaborative autonomy, the robot would have missed the door handle, which would have required a more involved operator intervention.

The behavior, still in `High` autonomy, executed the "Turn Clockwise" affordance of the door template. Once the affordance execution was finished, the behavior asked whether it should proceed with pushing the door or turn the handle more. The primary operator once again noticed a misalignment of the robot's hand and proceeded to adjust it using teleoperation. Then, after having communicated with the other two operators, the supervisor had the behavior repeat the turning part. The robot proceeded to turn the handle, but the motion was still not sufficient for unlatching the door; once again the supervisor had the behavior repeat the turning part. The behavior's response took longer than expected and so the primary operator completed the turning motion using the affordances of the template. The door handle unlatched and the door opened on its own (due to gravity). Thus, the supervisor had the behavior skip the execution of the "Push" affordance. Figure 6.17c shows a timeline of the events required to perform the task.

6.3 Laboratory Experimentation

This section describes lab experiments performed to demonstrate the potential of the contributions presented in this thesis. During the DRC Finals, hardware and communication issues prevented Team ViGIR from performing at the competitive level that this approach allows. For this reason, laboratory experimentation was performed using the same software setup as used during the DRC (with the exception of using the communications bridge). These tests were performed to compare a pure-operator execution of the task against a collaborative execution between high-level behaviors and operators. Given hardware issues after the DRC, the robot was not able to walk, for this reason lab experimentation does not include locomotion.

Opening the Door (Operator Only)

Laboratory experimentation of opening the door was performed to demonstrate that the system is capable of performing this task at an affordance level, as opposed to teleoperation as during day one of the DRC. In this case, the robot was already placed in a position where the door handle was reachable by the robot. The Door Template was inserted and aligned to the sensor data. The operator visualized the previews of the pre-grasp pose and the grasp pose to verify that the robot was able to reach both poses. Then, the operator commanded the robot to execute the arm motions for the pre-grasp pose. After reaching the pre-grasp pose, the operator needed to request the robot to set the fingers in a grasp posture before approaching the door handle. The operator then commanded the robot to move the arm to the grasp pose and setted the grasp posture that made the fingers to grasp the door handle. Once the robot had control of the door handle, the operator executed the “Turn Counter-clockwise” affordance of the Door template and the door got unlatched. Afterwards, the operator executes the “Push” affordance and the door was opened. Figure 6.17d shows a timeline of the events required to perform the task.

Opening the Door (High-level Behavior)

The same experiment as in Section 6.3 was also performed by a high-level behavior — monitored by the supervisor and in collaboration with the auxiliary operator. First, the behavior requests that the auxiliary operator inserts and aligns the door template. Once the template is in place, the behavior, executing in `High` autonomy level, is able to carry out all the remaining actions (pre-grasp, grasp, execution of affordances, etc.). The supervisor operator only had to confirm the few state machine transitions that had an autonomy threshold of `High` or `Full`. Figure 6.14 shows the series of images from the robot’s camera view of the execution of the task. Figure 6.17e shows a timeline of the events required to perform the task and Table 6.6 indicates the exact task completion time. The high-level behavior, in collaboration with the auxiliary operator, opened the door twice as fast as the primary operator acting alone.

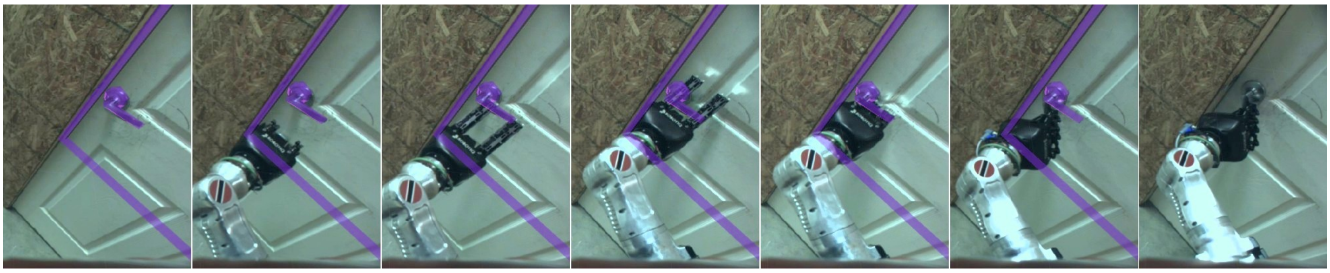


Figure 6.14: Robot’s camera view from door task. From left to right: Door template aligned, Pre-grasp, Open fingers, Grasp, Close fingers, Open Clockwise affordance, Push affordance.

Turning the Valve (Operator Only)

For this task, the robot was placed in a position where the valve was reachable so that locomotion was not required. The operator inserted the valve template and aligned it to the sensor data. Afterwards, the operator selected the visualization of the pre-grasp pose and initiated execution of the arm motion. Then, the same procedure was done to reach the grasp pose, which put the wrist attachment of the left hand in-between the crossbars of the valve. Finally, the operator selected the “Close” affordance of the valve template to generate the circular arm motions to turn the valve. Figure 6.15 shows the series of images from the robot’s camera view of the execution of the task. Figure 6.17f shows a timeline of the events required to perform the task.

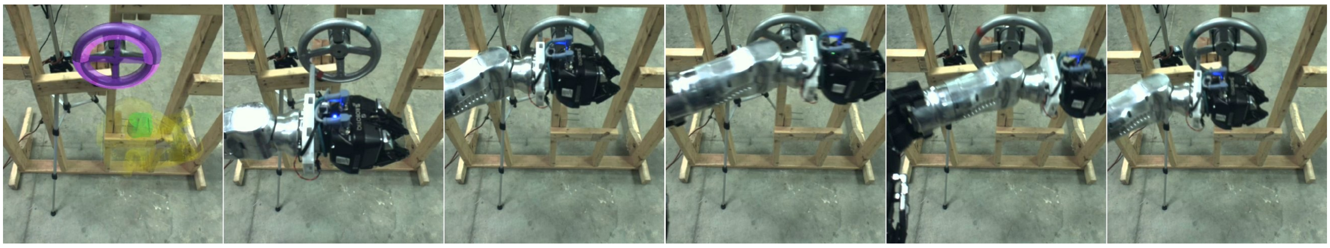
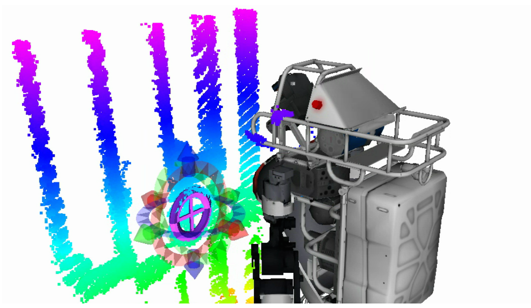


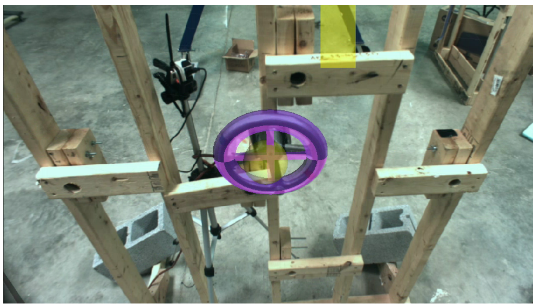
Figure 6.15: Robot's camera view from valve task. From left to right: Valve template aligned, Pre-grasp, Grasp, Open affordance 90, 180, and 270 degrees.

Turning the Valve (High-level Behavior)

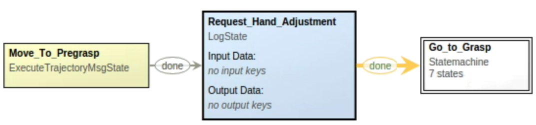
The same experiment as in Section 6.3 was also performed by a high-level behavior — monitored by the supervisor and in collaboration with the auxiliary and primary operators. First, the behavior requests that the auxiliary operator inserts and aligns the valve template. Once the template is in place, the behavior, executing in `High` autonomy level, moves the robot's arm to the pre-grasp pose. It then asks the primary operator to (optionally) adjust the hand's position to ensure that the wrist attachment will be able to slide into the valve. The primary operator responded that no adjustment was necessary. Thus, the supervisor allowed behavior execution to continue. Afterwards, the behavior was able to complete the task mostly autonomously; the supervisor made a few transition confirmations and high-level decisions.



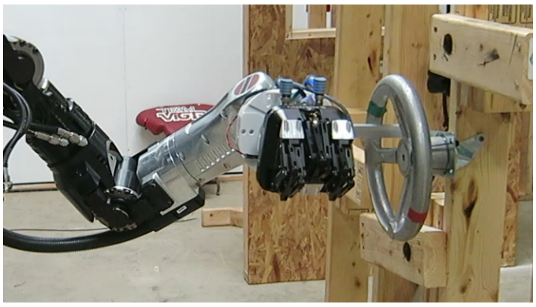
(a) Behavior requests Valve Template



(b) Valve Template alignment.



(c) Behavior asks for confirmation of Valve Template pose.



(d) Behavior inserts the wrist attachment and turn the valve.

Figure 6.16: "Turn valve" behavior execution in `High` autonomy level (i.e., most transitions do not require the supervisor's permission). The auxiliary operator placed the valve object template (top). The primary operator confirmed that the wrist attachment was aligned with the gaps in the valve (bottom).

Highlights from this experiment are depicted and discussed in Figure 6.16. Figure 6.17g shows a timeline of the events required to perform the task and Table 6.6 indicates the exact completion time. The high-level behavior, in collaboration with the auxiliary operator, turned the valve 150% faster compared to the primary operator acting alone.

6.3.1 Timeline Results Analysis

Figure 6.17 shows a timeline of the events that happened during the DRC Finals runs and the laboratory experimentation. In this timeline the periods of time where the robot was stepping and when manipulation was being performed can be appreciated. To fairly compare times, time starts after the robot has stopped locomotion. This timeline mainly shows when the robot was running under high-level behaviors and when it was being commanded by the human operator. Tasks that were run under a high-level behavior start with a request from the behavior (shown in yellow) to the auxiliary operator to provided the template ID of the aligned template (shown in lightgreen). The auxiliary operator responds to this request (shown in purple), and the behavior continues execution. The supervisor operator monitors the autonomy of the high-level behavior (shown in gray) and can execute priority manual transitions (shown in black).

Detailed timeline description is given only for the events during the Door task on day two (see Figure 6.17c), the other tasks can be analogously analysed. During the second day, after accomplishing the drive task, a reset was requested to skip the egress task. This reset consisted of a 10 min pause; in the meanwhile, the operators prepared the high-level behavior (shown in pink). After the reset pause was over, the supervisor gave a manual transition to the behavior, and the robot started stepping (shown in blue). The robot autonomously walked to the stand pose of the door template and rotated the torso towards the door (upper body motions including manipulation are shown in darkgreen). The supervisor operator changed the autonomy level to `High` so only transitions with equal or higher autonomy level were required to be confirmed. Then, the behavior reminded the auxiliary operator to consider template alignment; the auxiliary operator responded and the supervisor operator confirmed the transition. The behavior autonomously commanded the arm motions to pre-grasp and grasp pose and once again, the behavior requested for a confirmation if the robot's hand was in the correct pose. The primary operator noticed a misalignment and started generating commands using object templates (shown in cyan) to command the robot's hand to be closer to the door's handle using the "push" affordance of the door template. Then, the supervisor operator confirmed the transition and the robot started turning the door handle. The behavior requested for confirmation to continue turning the door handle or to start pushing the door. However, the primary operator noticed again a misalignment and proceeded with Cartesian teleoperation (shown in red) to adjust the robot's hand pose. Then, the supervisor confirmed the transition and the behavior continued turning the door handle. The behavior requested confirmation to continue turning the door handle or to start pushing the door; the supervisor confirmed to continue turning the door handle. However, the robot did not react to this command and the primary operator executed the step using the affordance of the door template and the door opened.

As a general view, gray color means that the task was performed using high-level behaviors, which gets reflected in minimal operator input. Whereas tasks that present red color, show a lot of teleoperation from the primary operator, meaning less autonomy. Table 6.6 shows the manipulation times required to perform the task.

Video footage of the DRC runs and laboratory experiments can be seen here [108].

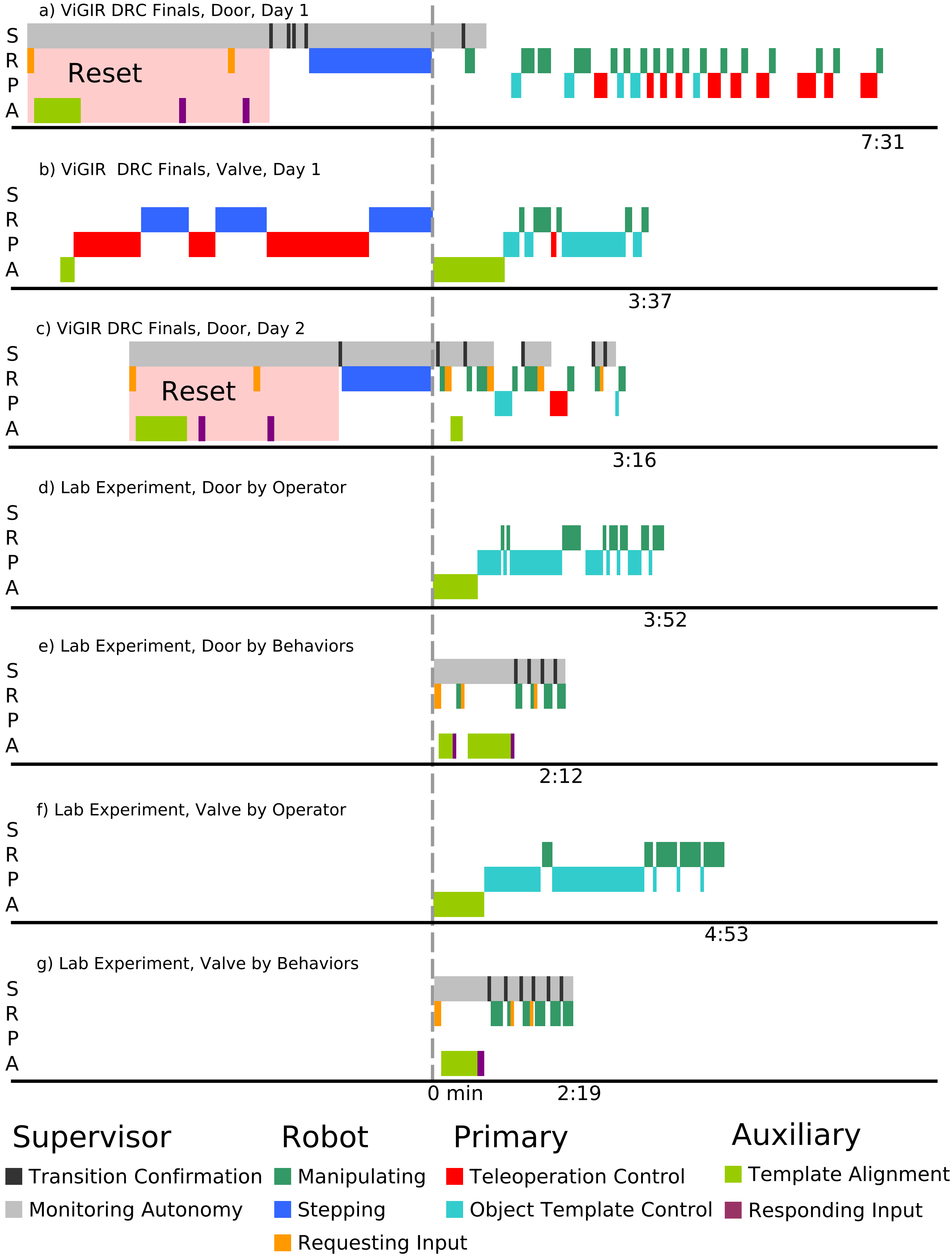


Figure 6.17: Timeline of actions during DRC Finals tasks and laboratory experimentation.

Task Times	Overall Time	Template Alignment	Teleoperation Control	Object Template Control	High-level Behaviors	Robot Motion
Door Day 1	07:31	(in Reset)	01:53	00:43	00:28	02:20
Lab Valve Op.	04:53	00:51	00:00	02:40	00:00	01:21
Lab Door Op.	03:52	00:45	00:00	01:57	00:00	01:00
Valve Day 1	03:37	01:12	00:05	01:38	00:00	00:42
Door Day 2	03:16	00:12	00:17	00:21	01:48	01:00
Lab Valve Beh.	02:19	00:36	00:00	00:00	02:19	00:53
Lab Door Beh.	02:12	00:57	00:00	00:00	02:12	00:31

Table 6.6: Time table of DRC tasks and laboratory evaluation. Due to operators analysing situation, and behaviors and robot motions being simultaneously executed, the overall time does not reflect the sum of the times taken for each of the these sub-tasks.

6.3.2 Experiments for Manipulation Skills Transferring

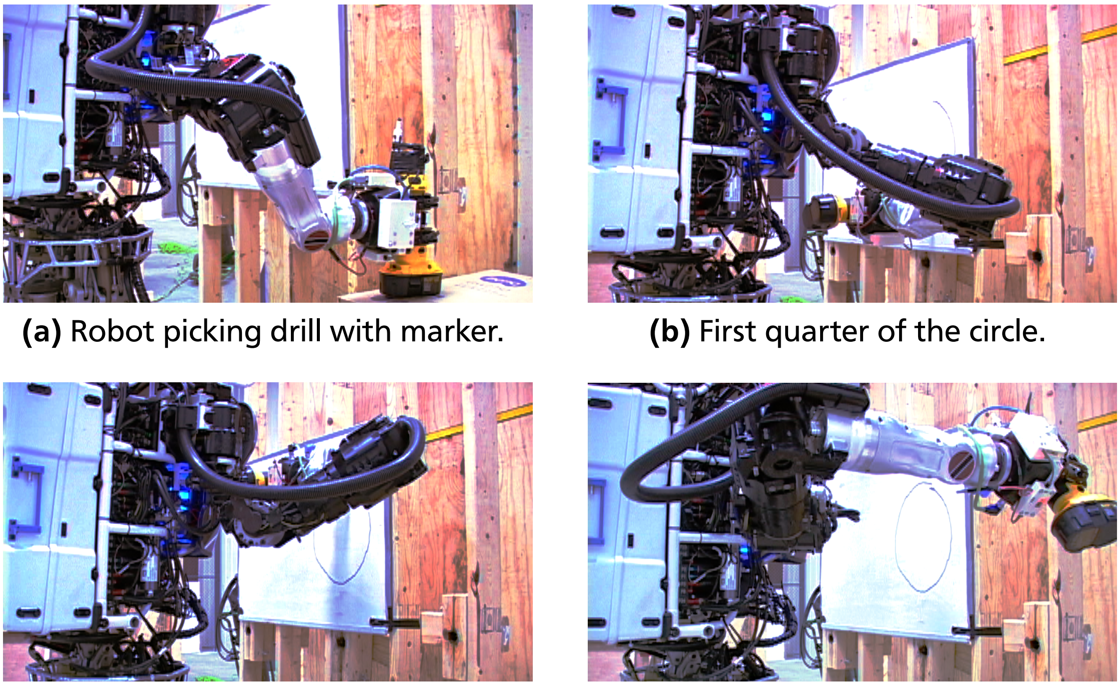
This section shows experiments that demonstrate how the developed object template approach enables to transfer affordances from one object to another or manipulating objects in a way they have not been used before can increase the potential to achieve a manipulation task. Experiments of type 1 will be used to demonstrate that the basic manipulation skills can be performed by the humanoid robot, also shown in [76]. Experiments of manipulation skill transferring types 2 and 3 directly represent the proposed approach in this thesis.

Drawing a Circle with Cutting Tool affordances: Type 1

This experiment shows a test designed to evaluate the necessary manipulation skills that will be required to cut a circle from a dry wall using a vertical drill. To test these motions, a black marker was installed instead of the drill bit, and used a white board to show the path that the drill is following. The manipulation class required to cut a circle in a drywall using a vertical drill is "ttr" which means that the drill should always be perpendicular to the drywall plane and that can be translated in upwards and downwards directions. The affordances of the vertical drill can then be used to draw a circle in a whiteboard since both manipulation tasks belong to the same class as shown in Figure 6.18.

Steering Wheel with turning Valve affordances: Type 2

In this experiment a human-robot team has been prepared for tasks involving valve manipulation (opening and closing). But in this case, the robot runs into a situation where using a car is needed and the robot has no previous knowledge of how to manipulate a steering wheel. The steering wheel is a constrained object of class "rxx", it can only rotate around one axis. Assuming the robot can fits in the driver seat of the vehicle, the operator can then utilize the valve template to operate the steering wheel. Since the affordances of the valve template can produce the same necessary movements to turn a steering wheel, the operator can overlap the valve template with the sensor data that belongs to the steering wheel and use the turn affordance of the valve template as shown in Figure 6.19.



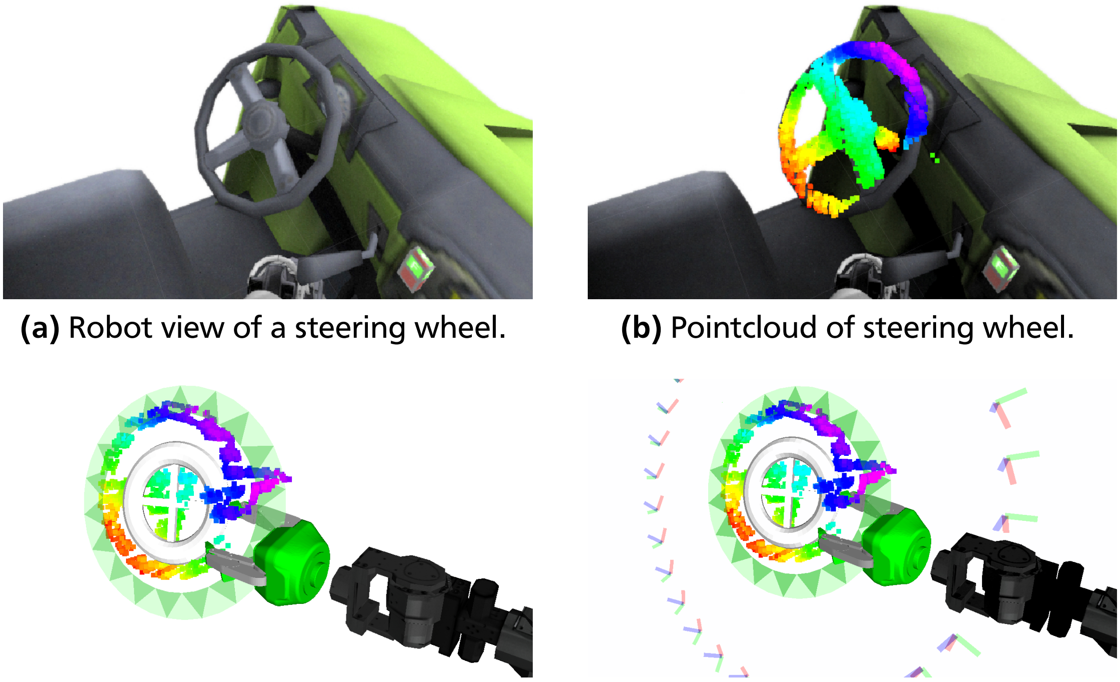
(a) Robot picking drill with marker.

(b) First quarter of the circle.

(c) Three quarters of the circle.

(d) Completed drawing circle.

Figure 6.18: The robot generating a clock-wise circular path to draw a circle in a whiteboard, but using the "cut circle" affordance of a vertical drill.



(a) Robot view of a steering wheel.

(b) Pointcloud of steering wheel.

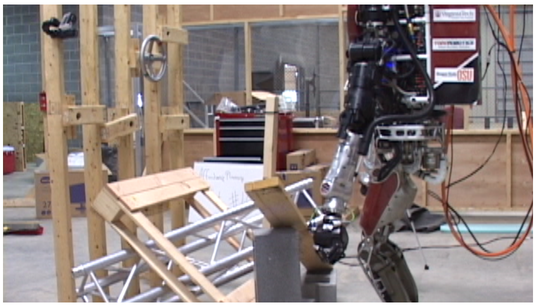
(c) Valve template overlapped with steering wheel pointcloud.

(d) Circular path of the wrist.

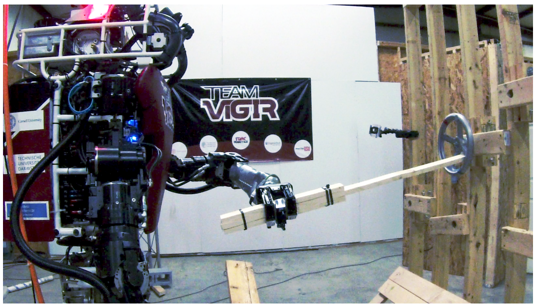
Figure 6.19: The robot generating a circular path to turn the steering wheel with the right hand, but utilizing the "turning" affordance of a valve template.

Reachable and turning a Valve with intermediary Objects: Type 3

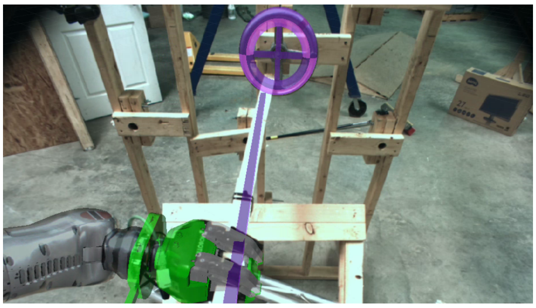
This experiment shows a situation where the robot needs to turn a valve, but in this case the valve is unreachable by the robot as shown in Figure 6.20. The human supervisor then finds a wooden stick which can be used to reach the valve as shown in Figure 6.20b. Since the affordances designed in this approach are grasp agnostic, the operator can easily command the robot to generate a circular path to rotate the stick around the axis of rotation of the valve, as described in Section 4.6. This experiment shows kinematic waypoints can be created on the fly regardless of the grasp pose with respect to the object and generate the same type of manipulation class required to accomplish the task.



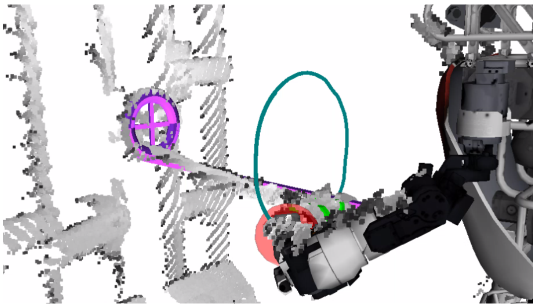
(a) Valve blocked by debris.



(b) Reaching valve with stick.



(c) Robot view from scene.



(d) Circular path of the hand.

Figure 6.20: The robot using the "turning" affordance of a valve template but utilizing a stick to reach the valve. The circular path shown in (d) is calculated for the hand with respect to the valve axis of rotation and keeping the end-effector orientation.

6.3.3 Experiments with Object Template Usabilities

A condensed version of this Section was published in: KI - Künstliche Intelligenz 2016 [78].

In this section selected experimental results are reported as proof of concept to demonstrate how using the object template approach allows to solve manipulation tasks in a versatile way. For example, being able to select points of interest in a grasped object can increase the potential of achieving a manipulation task by using specific points of interest in the grasped objects and extending the reachable workspace of a robot.

Drawing a Circle with the "Tip" Marker usability

The first experiment demonstrates the theoretical grounding of the approach by analysing the pattern followed by a usability with respect to an affordance. A Board Marker Template is created with a usability "tip" located at the painting edge of the board maker. The board marker is grasped by the robot and a Wall Template is used to create circles in a white board. Since the board marker is grasped with the finger tips as shown in Figure 6.21, the operator needs to align the Board Marker Template with the real object.

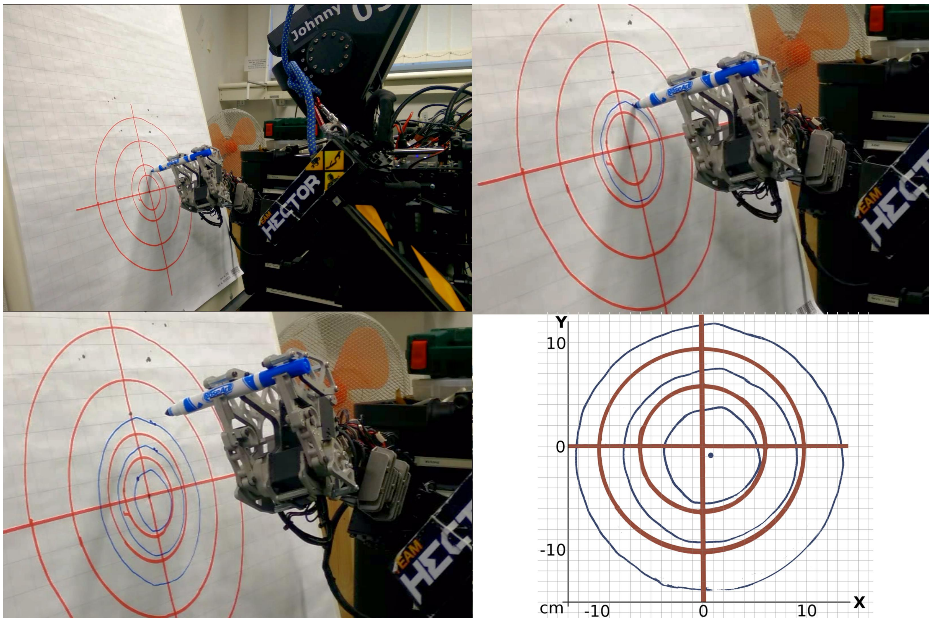


Figure 6.21: Johnny grasps a board marker with the finger tips (top left) and draws circles on the target board (top right). Three blue circles with different radius are drawn using the usability in the board marker template and the circular affordance of the wall template (bottom left). A digitalization of the circles drawn by the robot is made to show the pattern results (bottom right).

After alignment is complete, the operator attaches the Board Marker Template to the right hand and selects the usability "tip" from the user interface. The kinematic chain of the end-effector gets updated with the new transformations. Now that the Board Marker Template is attached, the operator can move the template to an initial position for drawing the circle through the user interface. With the marker in the initial pose, the operator can request a circular affordance from the Wall Template that will execute the necessary joint motions to move the board marker in a circular pattern around the center target of the Wall Template. From the digitalization made from the circles drawn by the robot, it can be seen that the three blue circles share the same center. However, inaccuracies from the manual alignment of the Board Marker Template from the human operator generates an error of 1.1 cm in the resulting pattern. For the purposes of the approach, these inaccuracies are not considered significant given that the tasks of the robot do not require high precision manipulation. The complete process of operator alignment and use of interface can be seen in this video [82].

Using a Drill and planing with respect to the "Bit" usability

The second experiment shows how using object usabilities, a proper motion pattern can be generated while planning with respect to specific points of interest in tools. In this example, the robot needs to use a tool (e.g. a drill) to cut out a circular pattern in a dry wall of around 20 cm diameter. When using the Drill Template, the nominal grasp is located around the grip of the drill, making the bit to be located 9 cm above the frame of reference of the end-effector. To draw a circle where the operator has planned, the drill bit needs to rotate around the center of axis rotation of the Wall Template. The bit usability will provide the motion planner with the right transformation to generate this pattern as seen in Figure 6.22b. An example of the transformations generated for this experiment can be seen in this video[84]. Since this transformation is calculated online, the drill can have any arbitrary orientation, in this case it is rotated ninety degrees compared to the orientation of the drill shown in Figure 4.9. The experiment is performed placing a board marker in the place of the bit and painting a circle in a white board in order to observe the generated pattern.

(a) (b)

Figure 6.22: Atlas first person view of the draw a circle in the wall task. An ovoid shape pattern can be appreciated which is generated due to joint controller inaccuracies generated by the drill mass, which is around 1.3*Kg*.

Executing manipulation outside the workspace of the robot

The third experiment is performed to demonstrate how the human operator can command the robot to use an object as an online-augmented end-effector. In this experiment, the robot is required to turn a valve, however, the robot is unable to do this without the use of a tool because the valve is in a higher place than the robot can reach as shown in Figure 6.23.

For this experiment, a long L-shaped stick (in this case a paint roller) which can be grasped and used to reach the valve is provided. The length of the paint roller was fixed, however, the precise total length is not relevant as long as the distance between the point where the robot grasps the object and the “roller” part is sufficient to reach the valve. This distance is automatically considered in the kinematic transformations after the operator requests that the object (the paint roller) gets attached to the end-effector (the hand).

The human operator identifies and commands the robot to grasp the paint roller. Once the robot has grasped the object, the human operator adapts and validates the alignment of the Paint Roller Template to match the pose of the real object in the robot’s hand. To be able to

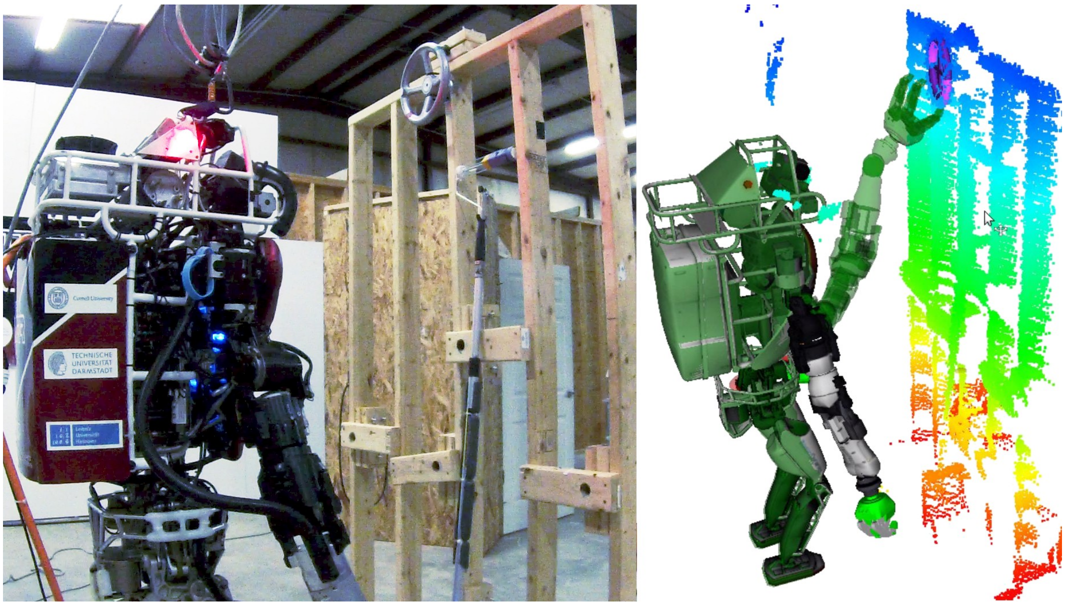


Figure 6.23: OCS view of the experiment setup. The operator has requested point cloud data of the environment and the Valve Template (purple) has been located to match the sensor data of the real valve. Atlas is unable to reach the valve as shown by the green ghost robot used for previewing the target arm motions.

turn the valve, the point that needs to follow a circular path around the axis of the valve is not located in the robot's hand but in the "roller" part of the object. To plan with respect to this point of interest in the grasped object, the operator can select the *usability* that belongs to that point (in this case the "roller usability"). The human operator can then command the robot to execute the turning affordance of the valve with this online-augmented end-effector as shown in Figure 6.24. This video[83] shows the complete process of grasping and manipulating the paint roller using its usabilities to rotate the valve.

6.4 Experiments from other Research Group

The object template approach presented in this thesis has also been applied by other research group, particularly the Terrestrial Robotics Engineering and Controls (TREC) laboratory in Virginia Tech, USA. This group also participated in the DRC Finals as Team VALOR[113] with the self-made robot Electric Series Compliant Humanoid for Emergency Response (ESCHER) [51] shown in Figure 6.25a. The team decided to not attempt the driving task and together with the Atlas robot from team Hong Kong University (HKU) were the only ones that successfully walked the complete distance from the start point up to the door. The attempt at opening the door was not successful due to encountered hardware issues. Footage from the attempt at the door during the DRC Finals can be seen in Figure 6.25b and Figure 6.25c. However, additional laboratory experiments performed after the DRC demonstrated that the object template approach was successfully applied to open a door, as can be seen in Figure 6.26. Another experiment was performed using a lever switch to deactivate an electric box as can be seen in Figure 6.27.

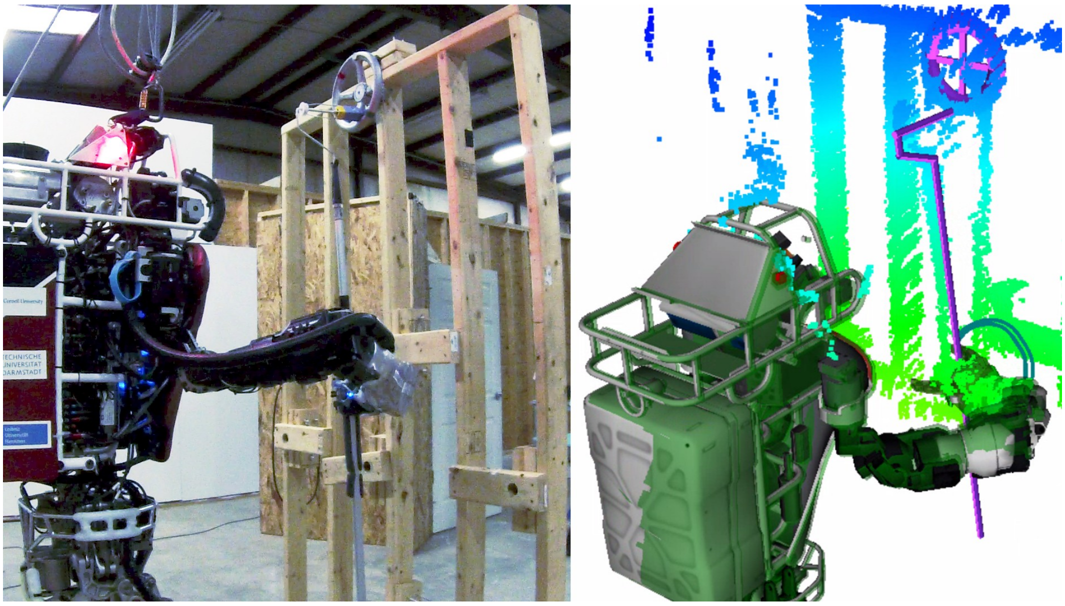
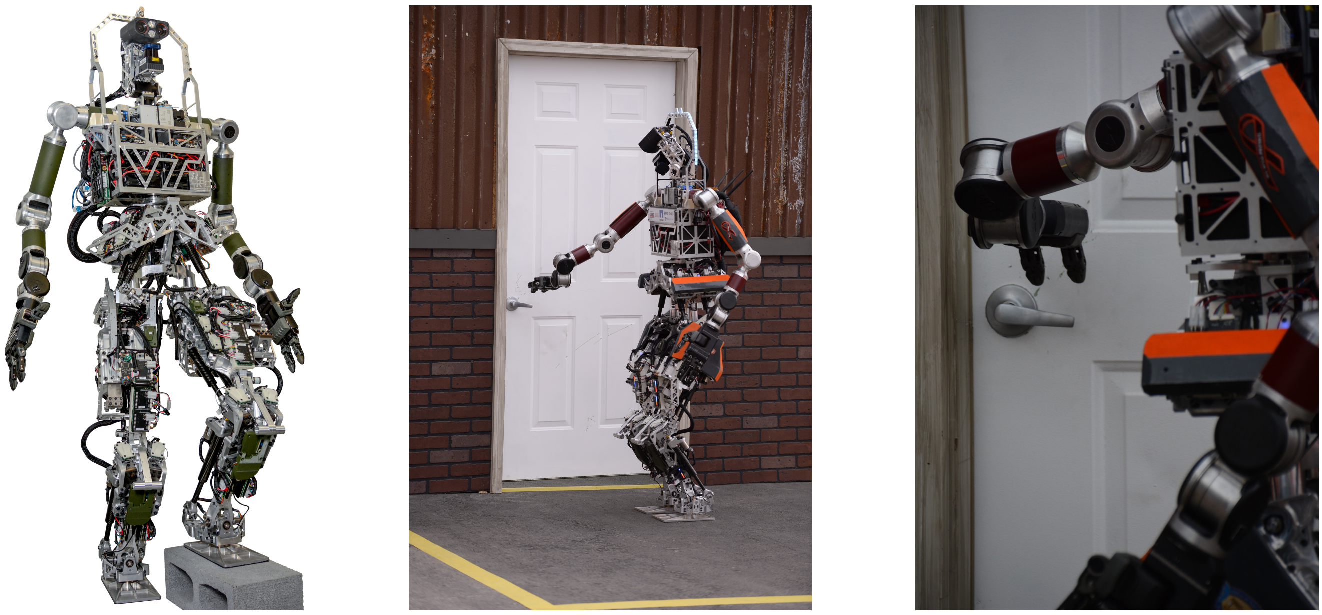


Figure 6.24: Atlas turning a high, non-arm-reachable valve using a "paint roller" as online-augmented end-effector. Atlas grasping the paint roller and inserting it between the valve cross-bars (left). OCS view of the experiment setup and the circular path performed by the robot's hand shown in dark green (right).



(a) (b) (c)

Figure 6.25: a) The ESCHER robot (image courtesy of Team Valor). b) Door opening attempt at the DRC Finals Door task. c) Close-up of the pre-grasp pose for the door handle. Images courtesy of DARPA.

Figure 6.26: The ESCHER robot opening the door using the door object template in a lab experiment. From left-top to bottom-right: Door setup, Walking to stand pose of door template, Pre-grasp pose, Grasp posture, Grasp, Turn Clockwise affordance, Push affordance, Door opened. Images courtesy of Team VALOR.

Figure 6.27: The ESCHER robot pulling down an electric break using the electric box object template in a lab experiment. Top row: Electric box setup, Walking to stand pose of box template, Pre-grasp pose, and Grasp. Bottom row: Pull down affordance in four sequenced images. Images courtesy of Team VALOR.

7 Conclusions

7.1 Summary of the Contributions

The field of manipulation control of remote supervised semi-autonomous robots is of significant relevance towards challenging future applications such as disaster response and recovery and space exploration, among others. The need for development of high-level interaction abilities between human operators and remote robots for executing manipulation tasks has been made clear with natural and human-made disasters. This thesis makes several contributions towards research and development of a versatile interaction means for manipulation using remote robotic systems which are summarized in the following paragraphs.

An Object Template Concept for Remote Manipulation Control

In order to provide a human operator with a highly efficient means of interaction at a high-level of abstraction with a remote robot to perform manipulation tasks using information based on the potential use of the objects of interest, the concept of object templates is presented (Chapter 3).

By generalizing the basic motions required to manipulate objects, the presented concept adds to the current state-of-the-art approaches the possibility of describing object motions at the affordance-level. This enables a broader range of modalities for object manipulation, from teleoperating the robot at a Cartesian level, commanding executions by a human supervisor, to abstracting the information into higher system layers such as autonomous behaviors.

By considering additional physical and abstract object information, the presented concept also adds the possibility of anticipating the requirements of performing a manipulation task. This is relevant for providing information to the remote robot, from information about the physical properties of the object (e.g., shape and mass) to anticipate the actuator control requirements and prevent object collisions with the environment, to information about which particular part of the objects of interest is required to achieve a specific manipulation task. Related own publications are [44], [45], and [76].

Object Template Library as a Common Framework for Humans and Robots

In order to provide remote human-supervised robotic systems with the possibility to collaboratively interact with the environment efficiently at an affordance level, a framework for defining and implementing the object template concept based on three general independent-blocks of information is presented (Chapter 4).

The Object Library is designed to contain specific object information. This information is particular for the object it represents and describes the physical aspects of the objects as well as abstract aspects such as the affordances of the object. The affordance information is agnostic the grasp pose and type of end-effector used as well as agnostic to the robotic system. The framework has been made available as open source to allow research groups with different robotic hardware to describe the objects of interest to be used during manipulation.

The Grasp Pose Library is designed to contain grasp information related to the particular end-effector being used during manipulation. This is relevant to research groups because of the high diversity in robotic end-effectors. Particular aspects such as number of fingers and number of joints, as well as information about potential pre-grasp and final poses to approach to a particular object are provided. This grasp definitions allow a robot-agnostic manipulation scheme.

The Stand Pose Library is designed to provide potential positioning information related to the place where a robot should stand in order to reach the object. The incorporation of relevant approaches for autonomously finding this information is possible, however, there are still some open topics for research. For this reason, in the approach presented here, this information is found through an offline empirical analysis, which can then be preloaded and used during online operations. Related own publication in [76].

Allowing the Human-Robot Team to Improvise and Perform Highly Versatile Manipulation Tasks

In order to achieve a manipulation task in a remote, unstructured, and possibly degraded environment, it is sometimes required that tasks or subtasks must be performed in a different way. Or objects need to be used for a different purpose than they were designed to (Chapter 4). The object template concept presented in this thesis has been demonstrated to be capable of allowing the human-robot team to utilize human intelligence to improvise during a manipulation task. This is relevant because having the possibility to flexibly command a robot to use objects in different ways increases the potential of achieving a complex manipulation task. The idea of transferring manipulation skills between objects or between tasks of the same class is performed using object templates in three different ways. First, by allowing transferring manipulation skills between objects in which physical properties can differ, but that they can still be considered to be the same type of object. Second, by allowing transferring manipulation skills between different objects, but with same manipulation classes. And third, by allowing transferring manipulation skills through the use of intermediate objects. Related own publication in [77].

A Manipulation Interface for Commanding Tasks Using the Object Template Concept

In order to provide a human operator or a group of operators with an interface to interact with a remote robotic system to perform manipulation tasks at an affordance-level, a user interface for commanding actions at an affordance-level with object control interaction is presented (Chapter 5).

This open source interface, provides a human operator with different levels of control in a manipulation task, from low level end-effector joint control over a generalized concept of “open” and “close” end-effector to a more high-level interaction of commanding execution of tasks at an affordance level. This is relevant because during a remote manipulation task lower levels of control such as teleoperation can prevent a task from being performed in a reliable way. Having the possibility to simply select actions at an affordance level and let the remote robot to compute and execute the generated trajectories autonomously increases the efficiency and reduces the execution time of tasks which in case of a disaster scenario can be a matter of life. Additionally, the object template concept contributes to a collaborative autonomy approach which systematically coordinate tasks between the human supervisor and the avatar robot. Related own publication in [79].

Experimental Evaluation

Evaluation of the performance of the approach through experimentation is required to validate the presented contributions (Chapter 6). System-oriented evaluations and performance comparison with other state-of-the-art approaches have been performed through participation in the renowned international competition for disaster response "DARPA Robotics Challenge" (Section 6.2). Systematic laboratory experimentations were performed to validate the approach focusing on specific capabilities or aspects without considering the challenges added from a robotics competition (Section 6.3).

The presented approach has also been tested and experimentally validated by another research group in the USA (Section 6.4). This demonstrates the capability of the presented approach to be used by third-parties and with different robot hardware systems.

7.2 Outlook

The contributions presented in this thesis provide a new approach to efficiently perform remote manipulation tasks using remote avatar robots under human supervision. However, there are still open topics that require further research.

Increasing perception capabilities of robots has to be achieved for situations where communication with the human operator is lost. The presented approach relies on the capability of a human operator perceiving the remote environment using the (preprocessed) sensor data provided by the robot. When this limitation is overcome and robots can autonomously identify known as well as unknown objects in unstructured and degraded environments [88], the object template concept continues to serve as basis for object manipulation.

3D interaction is not yet in a perfect state. Manipulating object templates in a virtual environment with peripherals such as mouse, keyboard, or more advanced interaction systems such as 6DOF joysticks and 3D vision systems still requires training and is not yet reliable enough to be used during real world disasters by first responders.

Object manipulation capabilities in general need to be improved further. For example, autonomous grasping [24] and bi-manual manipulation of an object, are not yet considered in the presented approach. Scaling of the object templates is also a limitation of the current implementation; providing the possibility to change the scale of object templates will increase the flexibility of the approach to be adapted to objects of different size on the fly. The object template information could also be in principle extended to consider how objects are required to be grasped using multiple end-effectors, as well as considering the forces that are required to be applied on an object in order to achieve a manipulation task instead of just considering the kinematic constraints of the task.

Bibliography

[1] Ackerman, Evan. Astronaut Aboard the ISS Controls a Robot on Earth Using Haptic Feedback. `http://spectrum.ieee.org/automaton/robotics/space-robots/esa-space-teleoperation-tests`, 2015. Accessed: 2015-12-17.

[2] T. Asfour, P. Azad, N. Vahrenkamp, K. Regenstein, A. Bierbaum, K. Welke, J. Schröder, and R. Dillmann. Toward Humanoid Manipulation in Human-Centred Environments. *Robotics and Autonomous Systems*, 56:54–65, 2008.

[3] T. Asfour, K. Regenstein, P. Azad, J. Schröder, N. Vahrenkamp, and R. Dillmann. ARMAR-III: An Integrated Humanoid Platform for Sensory-Motor Control. In *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, pages 169–175, 2006.

[4] N. Banerjee, X. Long, R. Du, F. Polido, S. Feng, C. G. Atkeson, M. Gennert, and T. Padir. Human-Supervised Control of the ATLAS Humanoid Robot for Traversing Doors. In *Humanoid Robots (Humanoids), 15th IEEE-RAS International Conference on*, 2015.

[5] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mosenlechner, D. Pangercic, T. Ruhr, and M. Tenorth. Robotic roommates making pancakes. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 529–536. IEEE, 2011.

[6] A. Bierbaum and M. Rambow. Grasp Affordances from Multi-fingered Tactile Exploration using Dynamic Potential Fields. In *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 168–174. IEEE, 2009.

[7] P. Birkenkampf, D. Leidner, and C. Borst. A knowledge-driven shared autonomy human-robot interface for tablet computers. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 152–159, Nov 2014.

[8] N. Blodow, L. C. Goron, Z.-C. Marton, D. Pangercic, T. Ruhr, M. Tenorth, and M. Beetz. Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4263–4270. IEEE, 2011.

[9] J. Bradshaw, P. Feltovich, and M. Johnson. *Handbook of Human-Machine Interaction*, chapter Human-Agent Interaction, pages 283–302. Ashgate, 2011.

[10] J. Brookshire, S. Singh, and R. Simmons. Preliminary Results in Sliding Autonomy for Assembly by Coordinated Teams. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 1, pages 706–711 vol.1, Sept 2004.

[11] A. Brygo, I. Sarakoglou, N. Tsagarakis, and D. Caldwell. Tele-manipulation with a humanoid robot under autonomous joint impedance regulation and vibrotactile balancing feedback. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 862–867, Nov 2014.

[12] I. Bullock and A. Dollar. Classifying human manipulation behavior. In *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*, pages 1–6, June 2011.

[13] F. Burget and M. Bennewitz. Stance selection for humanoid grasping tasks by inverse reachability maps. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 5669–5674, May 2015.

[14] G. Cheng and A. Zelinsky. Supervised autonomy: A framework for human-robot systems development. *Autonomous Robots*, 10(3):251–266, 2001.

[15] S. Chitta, I. Sucan, and S. Cousins. Moveit! *IEEE Robotics Automation Magazine*, 19(1):18–19, 2012.

[16] D. Cong, K.-H. Lee, and W. Newman. Manipulation planning for the Atlas humanoid robot. In *Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on*, pages 1118–1123, Dec 2014.

[17] J. Crandall and M. Goodrich. Experiments in adjustable autonomy. In *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, volume 3, pages 1624–1629 vol.3, 2001.

[18] DARPA. The DARPA Robotics Challenge. `http://www.theroboticschallenge.org/`, 2012. Accessed: 2015-12-03.

[19] DARPA Robotics Challenge Finals. Team ViGIR Time Lapse 1 - Day 1. `https://www.youtube.com/watch?v=VEsUICAa4rg`, 2015. Accessed: 2015-12-03.

[20] DARPA Robotics Challenge Trials. DRC Trials Blue Hose Task Day 1. `https://www.youtube.com/watch?v=n3aGCe0vDLU`, 2013. Accessed: 2014-10-11.

[21] DARPA Robotics Challenge Trials. DRC Trials Blue Hose Task Day 2. `https://www.youtube.com/watch?v=B35A7-JDIdI`, 2013. Accessed: 2014-10-11.

[22] M. R. Endsley. Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64, 1995.

[23] M. Fallon, S. Kuindersma, S. Karumanchi, M. Antone, T. Schneider, H. Dai, C. Perez D'Arpino, R. Deits, M. DiCicco, D. Fourie, et al. An architecture for online affordance-based perception and whole-body planning, 2014.

[24] D. R. Faria, R. Martins, J. Lobo, and J. Dias. Extracting data from human manipulation of objects towards improving autonomous robotic grasping. *Robotics and Autonomous Systems*, 60(3):396 – 410, 2012. Autonomous Grasping.

[25] T. Feix, I. Bullock, and A. Dollar. Analysis of Human Grasping Behavior: Correlating Tasks, Objects and Grasps. *Haptics, IEEE Transactions on*, 7(4):430–441, Oct 2014.

[26] R. E. Fikes and N. J. Nilsson. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. In *Proceedings of the 2Nd International Joint Conference on Artificial Intelligence*, IJCAI'71, pages 608–620, San Francisco, CA, USA, 1971. Morgan Kaufmann Publishers Inc.

[27] T. W. Fong and I. Nourbakhsh. Peer-to-Peer Human-Robot Interaction for Space Exploration. In *AAAI Fall Symposium*. AAAI, October 2004.

[28] T. W. Fong, C. Thorpe, and C. Baur. Collaborative Control: A Robot-Centric Model for Vehicle Teleoperation. In *AAAI 1999 Spring Symposium: Agents with Adjustable Autonomy*, March 1999.

[29] Future Robotics Technology Center. The Fukushima nuclear power plant reactor building in the investigation by (TEPCO). `https://www.youtube.com/watch?v=JBkJjIFwSqU`, Jul 2011. Accessed: 2015-12-17.

[30] J. J. Gibson. The theory of affordances. In R. Shaw and E. J. Bransford, editors, *Perceiving, Acting, and Knowing: Toward an Ecological Psychology*, pages 67–82, Hilldale, USA, 1977.

[31] A. Goncalves, G. Saponaro, L. Jamone, and A. Bernardino. Learning visual affordances of objects and tools through autonomous robot exploration. In *Autonomous Robot Systems and Competitions (ICARSC), 2014 IEEE International Conference on*, pages 128–133, May 2014.

[32] M. A. Goodrich, J. W. Crandall, and E. Barakova. Teleoperation and beyond for assistive humanoid robots. *Reviews of Human Factors and Ergonomics*, 9(1):175–226, 2013.

[33] S. Hart, P. Dinh, and K. Hambuchen. The Affordance Template ROS Package for Robot Task Programming. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, 2015.

[34] S. S. Heikkilä, A. Halme, and A. Schiele. Affordance-based indirect task communication for astronaut-robot cooperation. *Journal of Field Robotics*, 29(4):576–600, 2012.

[35] A. Hornung, A. Dornbush, M. Likhachev, and M. Bennewitz. Anytime search-based footstep planning with suboptimality bounds. In *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, pages 674–679. IEEE, 2012.

[36] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees. *Autonomous Robots*, 2013. Software available at `http://octomap.github.com`.

[37] R. Jain and T. Inamura. Learning of tool affordances for autonomous tool manipulation. In *System Integration (SII), 2011 IEEE/SICE International Symposium on*, pages 814–819, Dec 2011.

[38] M. Johnson, J. Bradshaw, P. Feltovich, C. Jonker, B. van Riemsdijk, and M. Sierhuis. The Fundamental Principle of Coactive Design: Interdependence Must Shape Autonomy. In M. De Vos, N. Fornara, J. Pitt, and G. Vouros, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems VI*, volume 6541 of *Lecture Notes in Computer Science*, pages 172–191. Springer Berlin Heidelberg, 2011.

[39] M. Johnson, B. Shrewsbury, S. Bertrand, T. Wu, D. Duran, M. Floyd, P. Abeles, D. Stephen, N. Mertins, A. Lesman, J. Carff, W. Rifenburgh, P. Kaveti, W. Straatman, J. Smith, M. Griffioen, B. Layton, T. de Boer, T. Koolen, P. Neuhaus, and J. Pratt. Team IHMC's Lessons Learned from the DARPA Robotics Challenge Trials. *Journal of Field Robotics*, 32(2):192–208, 2015.

[40] P. Kaiser, D. Gonzalez-Aguirre, F. Schültje, J. Borràs, N. Vahrenkamp, and T. Asfour. Extracting whole-body affordances from multimodal exploration. In *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, pages 1036–1043, 2014.

[41] P. Kaiser, N. Vahrenkamp, F. Schültje, J. Borràs, and T. Asfour. Extraction of Whole-Body Affordances for Loco-Manipulation Tasks. *International Journal of Humanoid Robotics*, 12(3):0–0, 2015.

[42] G. Klien, D. Woods, J. Bradshaw, R. Hoffman, and P. Feltovich. Ten challenges for making automation a "team player" in joint human-agent activity. *Intelligent Systems, IEEE*, 19(6):91–95, Nov 2004.

[43] S. Kohlbrecher. *A Holistic Approach for Highly Versatile Supervised Autonomous Urban Search and Rescue Robots*. PhD thesis, TU Darmstadt, Department of Computer Science, December 2 2015.

[44] S. Kohlbrecher, D. C. Conner, A. Romay, F. Bacim, D. A. Bowman, and O. von Stryk. Overview of Team ViGIR's approach to the Virtual Robotics Challenge. In *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*, pages 1–2. IEEE, 2013.

[45] S. Kohlbrecher, A. Romay, A. Stumpf, A. Gupta, O. von Stryk, F. Bacim, D. A. Bowman, A. Goins, R. Balasubramanian, and D. C. Conner. Human-robot Teaming for Rescue Missions: Team ViGIR's Approach to the 2013 DARPA Robotics Challenge Trials. *Journal of Field Robotics*, 32(3):352–377, 2015.

[46] S. Kohlbrecher, A. Stumpf, A. Romay, P. Schillinger, O. von Stryk, and D. C. Conner. A Comprehensive Software Framework for Complex Locomotion and Manipulation Tasks Applicable to Different Types of Humanoid Robots. *Frontiers in Robotics and AI*, 3:31, 2016.

[47] T. Koolen, J. Smith, G. Thomas, S. Bertrand, J. Carff, N. Mertins, D. Stephen, P. Abeles, J. Englsberger, S. Mccrory, et al. Summary of team IHMC's Virtual Robotics Challenge entry. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2013.

[48] D. Kortenkamp. Designing an Architecture for Adjustably Autonomous Robot Teams. In R. Kowalczyk, S. Loke, N. Reed, and G. Williams, editors, *Advances in Artificial Intelligence. PRICAI 2000 Workshop Reader*, volume 2112 of *Lecture Notes in Computer Science*, pages 335–338. Springer Berlin Heidelberg, 2001.

[49] O. Kroemer, C. Daniel, G. Neumann, H. van Hoof, and J. Peters. Towards learning hierarchical skills for multi-phase manipulation tasks. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2015.

[50] N. Krüger, C. W. Geib, J. H. Piater, R. P. A. Petrick, M. Steedman, F. Wörgötter, A. Ude, T. Asfour, D. Kraft, D. Omrcen, A. Agostini, and R. Dillmann. Object-action complexes: Grounded abstractions of sensory-motor processes. *Robotics and Autonomous Systems*, 59(10):740–757, 2011.

[51] B. K. T.-S. Lee. Design of a Humanoid Robot for Disaster Response. Master's thesis, Virginia Tech, 2014.

[52] A. Leeper, K. Hsiao, M. Ciocarlie, I. Sucan, and K. Salisbury. Methods for Collision-Free Arm Teleoperation in Clutter Using Constraints from 3D Sensor Data. In *IEEE Intl. Conf. on Humanoid Robots*, Atlanta, GA, 10/2013 2013.

[53] D. Leidner, C. Borst, and G. Hirzinger. Things are made for what they are: Solving manipulation tasks by using functional object classes. In *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, pages 429–435, Nov 2012.

[54] R. Lioutikov, O. Kroemer, J. Peters, and G. Maeda. Learning manipulation by sequencing motor primitives with a two-armed robot. In *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS)*, 2014.

[55] J. Liu, F. Feng, Y. C. Nakamura, and N. S. Pollard. A taxonomy of everyday grasps in action. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 573–580, Nov 2014.

[56] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins. PDDL—The Planning Domain Definition Language. Technical report, Yale Center for Computational Vision and Control, New Haven, CT, 1998.

[57] P. McKerrow. *Introduction to Robotics*. ACM Press frontier series. Addison-Wesley Publishing Company, 1991.

[58] A. T. Miller and P. K. Allen. GraspIt! a versatile simulator for robotic grasping. *Robotics & Automation Magazine, IEEE*, 11(4):110–122, 2004.

[59] R. Moratz and T. Tenbrink. Affordance-based human-robot interaction. In E. Rome, J. Hertzberg, and G. Dorffner, editors, *Towards Affordance-Based Robot Control*, volume 4760 of *Lecture Notes in Computer Science*, pages 63–76. Springer Berlin Heidelberg, 2008.

[60] J. Morrow and P. Khosla. Manipulation task primitives for composing robot skills. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 4, pages 3354–3359 vol.4, Apr 1997.

[61] MoveIt! Grasp Message. `http://docs.ros.org/indigo/api/moveit_msgs/html/msg/Grasp.html`, 2015. Accessed: 2015-12-18.

[62] MoveIt! Planning Scene. `http://moveit.ros.org/documentation/concepts/`, 2015. Accessed: 2015-12-17.

[63] R. R. Murphy. *Introduction to AI Robotics*. MIT Press, Cambridge, MA, USA, 2000.

[64] A. Myers, C. Teo, C. Fermuller, and Y. Aloimonos. Affordance detection of tool parts from geometric features. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1374–1381, May 2015.

[65] K. Nagatani, S. Kiribayashi, Y. Okada, K. Otake, K. Yoshida, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, M. Fukushima, et al. Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots. *Journal of Field Robotics*, 30(1):44–63, 2013.

[66] K. Nagatani, S. Kiribayashi, Y. Okada, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, and Y. Hada. Redesign of rescue mobile robot quince. In *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pages 13–18. IEEE, 2011.

[67] M. Nieuwenhuisen, J. Stückler, A. Berner, R. Klein, and S. Behnke. Shape-primitive based object recognition and grasping. In *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, pages 1–5. VDE, 2012.

[68] L. U. Odhner, L. P. Jentoft, M. R. Claffee, N. Corson, Y. Tenzer, R. R. Ma, M. Buehler, R. Kohout, R. D. Howe, and A. M. Dollar. A compliant, underactuated hand for robust manipulation. *International Journal of Robotics Research*, 2013.

[69] Open Source Robotics Foundation. Gazebo Simulator. `http://gazebosim.org/`, 2015. Accessed: 2015-12-03.

[70] K. Petersen. *General Concepts for Human Supervision of Autonomous Robot Teams*. PhD thesis, TU Darmstadt, Department of Computer Science, July 9 2013.

[71] J. Pratt. Towards Humanoid Avatar Robots for Co-Exploration of Hazardous Environments. `http://www.humanoids2014.com/index.php/program/plenary-talks`, 2014. Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on.

[72] M. Przybylski, N. Vahrenkamp, T. Asfour, and R. Dillmann. Grasp and Motion Planning for Humanoid Robots. In *Grasping in Robotics*, pages 329–359. Springer, 2013.

[73] J. Rebelo, T. Sednaoui, E. den Exter, T. Krueger, and A. Schiele. Bilateral robot teleoperation: A wearable arm exoskeleton featuring an intuitive user interface. *Robotics Automation Magazine, IEEE*, 21(4):62–69, Dec 2014.

[74] Robotiq. 3-finger adaptive robot gripper. `http://robotiq.com/products/industrial-robot-hand/`, 2016. Accessed: 2016-03-03.

[75] E. Rocon, A. F. Ruiz, J. L. Pons, L. Beccai, S. Micera, C. Cipriani, J. Carpaneto, M. C. Carrozza, A. Schiele, S. Roccella, E. Cattin, N. Vitiello, F. Vecchi, J. C. Perry, J. Rosen, N. G. Tsagarakis, D. G. Caldwell, and S. Kousidou. *Wearable Upper Limb Robots*, pages 235–281. John Wiley & Sons, Ltd, 2008.

[76] A. Romay, S. Kohlbrecher, D. C. Conner, A. Stumpf, and O. von Stryk. Template-based Manipulation in Unstructured Environments for Supervised Semi-autonomous Humanoid Robots. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 979–986, Nov 2014.

[77] A. Romay, S. Kohlbrecher, D. C. Conner, and O. von Stryk. Achieving Versatile Manipulation Tasks with Unknown Objects by Supervised Humanoid Robots based on Object Templates. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 249–255, Nov 2015.

[78] A. Romay, S. Kohlbrecher, and O. v. Stryk. An object template approach to manipulation for humanoid avatar robots for rescue tasks. *KI - Künstliche Intelligenz*, 30(3):279–287, 2016.

[79] A. Romay, S. Maniatopoulos, S. Kohlbrecher, A. Stumpf, O. von Stryk, H. Kress-Gazit, P. Schillinger, and D. C. Conner. Collaborative Autonomy between High-level Behaviors and Human Operators for Remote Manipulation Tasks using Different Humanoid Robots. *Journal of Field Robotics*, 2016.

[80] Romay, Alberto. VRC Task 3 part 1. https://www.youtube.com/watch?v=SYdpU4TCINU, 2013. Accessed: 2015-12-03.

[81] Romay, Alberto. Valve Turning with Template Based Manipulation Approach. https://www.youtube.com/watch?v=wKFJO-Zkjck, 2015. Accessed: 2015-12-17.

[82] Romay, Alberto. Drawing a Circle with the "Tip" Marker usability. https://www.youtube.com/watch?v=pv8bEQEx1yM&list=PLqdOEBv9QGrHrqOQCvdbkDDBHVlTOKJcL, 2016. Accessed: 2016-08-17.

[83] Romay, Alberto. Executing manipulation outside the workspace of the robot. https://www.youtube.com/watch?v=4km_aaatA0M, 2016. Accessed: 2016-08-17.

[84] Romay, Alberto. Using a Drill and planing with respect to the "Bit" usability. https://www.youtube.com/watch?v=BdeBzgJIPx0, 2016. Accessed: 2016-08-17.

[85] ROS. Rviz Interactive Markers. http://wiki.ros.org/rviz/Tutorials/Interactive%20Markers%3A%20Getting%20Started, 2015. Accessed: 2015-12-04.

[86] M. Rouleau and D. Hong. Design of an underactuated robotic end-effector with a focus on power tool manipulation. In *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V05BT08A027–V05BT08A027. American Society of Mechanical Engineers, 2014.

[87] M. T. Rouleau. *Design and Evaluation of an Underactuated Robotic Gripper for Manipulation Associated with Disaster Response*. PhD thesis, Virginia Tech, 2015.

[88] R. B. Rusu. *Semantic 3D Object Maps for Everyday Robot Manipulation*. Springer Publishing Company, Incorporated, 2013.

[89] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1–4. IEEE, 2011.

[90] E. Şahin, M. Çakmak, M. R. Doğar, E. Uğur, and G. Üçoluk. To afford or not to afford: A new formalization of affordances toward affordance-based robot control. *Adaptive Behavior*, 15(4):447–472, 2007.

[91] Sandia National Laboratories. Sandia hand. http://www.sandia.gov/research/robotics/advanced_manipulation/Sandia_Hand.html, 2014. Accessed: 2014-03-07.

[92] D. Schiebener, J. Schill, and T. Asfour. Discovery, Segmentation and Reactive Grasping of Unknown Objects. In *Humanoids*, pages 71–77, 2012.

[93] A. Schiele and G. Visentin. The ESA human arm exoskeleton for space robotics telepresence. *(iSAIRAS) 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, pages 19–23, 2003.

[94] P. Schillinger. An Approach for Runtime-Modifiable Behavior Control of Humanoid Rescue Robots. Master's thesis, Technische Universität Darmstadt, 2015.

[95] P. Schillinger, S. Kohlbrecher, and O. von Stryk. Human-robot collaborative high-level control with application to rescue robotics. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, page to appear, 2016.

[96] Schillinger, Philipp. Flexible Behavior Engine (FlexBe). `https://github.com/team-vigir/flexbe_behavior_engine`, 2015. Accessed: 2015-12-07.

[97] E. Schmidlin. *Can this robot pass through that aperture? Learning to make judgments for a tele-operated robot.* PhD thesis, Texas Tech University, 2014.

[98] K. Shoemake. Animating rotation with quaternion curves. *ACM SIGGRAPH computer graphics*, 19(3):245–254, 1985.

[99] J. Sinapov and A. Stoytchev. Learning and generalization of behavior-grounded tool affordances. In *Development and Learning, 2007. ICDL 2007. IEEE 6th International Conference on*, pages 19–24, July 2007.

[100] M. Stilman, M. Zafar, C. Erdogan, P. Hou, S. Reynolds-Haertle, and G. Tracy. Robots using environment objects as tools the "MacGyver" paradigm for mobile manipulation. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2568–2568, May 2014.

[101] A. Stoytchev. Behavior-grounded representation of tool affordances. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3060–3065, April 2005.

[102] A. Stoytchev. Learning the Affordances of Tools Using a Behavior-Grounded Approach. In E. Rome, J. Hertzberg, and G. Dorffner, editors, *Towards Affordance-Based Robot Control*, volume 4760 of *Lecture Notes in Computer Science*, pages 140–158. Springer Berlin Heidelberg, 2008.

[103] J. Stuckler and S. Behnke. Adaptive tool-use strategies for anthropomorphic service robots. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 755–760, Nov 2014.

[104] A. Stumpf, S. Kohlbrecher, D. C. Conner, and O. von Stryk. Supervised footstep planing for humanoid robots in rough terrain tasks using a black box walking controller. In *Humanoid Robots (Humanoids), 2014. 14th IEEE-RAS International Conference on*. IEEE, 2014.

[105] J. Sturm. Learning Kinematic Models of Articulated Objects. In *Approaches to Probabilistic Model Learning for Mobile Manipulation Robots*, volume 89 of *Springer Tracts in Advanced Robotics*, pages 65–111. Springer Berlin Heidelberg, 2013.

[106] I. Sucan. ROS URDF package, http://wiki.ros.org/urdf. online, September 2015. [Online; accessed September, 2015].

[107] Team Hector. Heterogeneous Cooperating Teams of Robots. `http://www.teamhector.de`, 2015. Accessed: 2016-03-08.

[108] Team ViGIR. DRC Finals Tasks and Laboriatory Experimentation. https://www.youtube.com/watch?v=5bSwwnQXfgQ&list=PL09J37oL4U7BXuuNHWE54vcMwsJlgAjmJ, 2015. Accessed: 2015-12-03.

[109] Team ViGIR. Virginia-Germany Interdiciplinary Robotics. http://www.teamvigir.org/, 2015. Accessed: 2015-12-03.

[110] R. Tedrake. Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems. http://drake.mit.edu, 2014. Accessed: 2015-12-07.

[111] A. ten Pas and R. Platt. Localizing Grasp Affordances in 3D Points Clouds Using Taubin Quadric Fitting. *CoRR*, abs/1311.3192, 2013.

[112] M. Tenorth, S. Profanter, F. Balint-Benczedi, and M. Beetz. Decomposing CAD models of objects of daily use and reasoning about their functional parts. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 5943–5949, Nov 2013.

[113] Terrestrial Robotics Engineering & Controls Lab. Team VALOR. http://www.me.vt.edu/trec/team-valor/, 2015. Accessed: 2015-12-03.

[114] V. Tikhanoff, U. Pattacini, L. Natale, and G. Metta. Exploring affordances and tool use on the iCub. In *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*, pages 130–137, Oct 2013.

[115] N. Vahrenkamp, T. Asfour, and R. Dillmann. Robot placement based on reachability inversion. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1970–1975. IEEE, 2013.

[116] N. Vahrenkamp, M. Kröhnert, S. Ulbrich, T. Asfour, G. Metta, R. Dillmann, and G. Sandini. Simox: A Robotics Toolbox for Simulation, Motion and Grasp Planning. In *International Conference on Intelligent Autonomous Systems (IAS)*, pages 585–594, 2012.

[117] K. Welke, J. Issac, D. Schiebener, T. Asfour, and R. Dillmann. Autonomous Acquisition of Visual Multi-view Object Representations for Object Recognition on a Humanoid Robot. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2012–2019. IEEE, 2010.

[118] J. M. Whetten, M. A. Goodrich, and Y. Guo. Beyond robot fan-out: Towards multi-operator supervisory control. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pages 2008–2015. IEEE, 2010.

[119] C. D. Wickens. Situation Awareness: Review of Mica Endsley's 1995 articles on Situation Awareness Theory and Measurement. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 50(3):397–403, 2008.

[120] S.-J. Yi, S. G. McGill, L. Vadakedathu, Q. He, I. Ha, J. Han, H. Song, M. Rouleau, B.-T. Zhang, D. Hong, M. Yim, and D. D. Lee. Team THOR's Entry in the DARPA Robotics Challenge Trials 2013. *Journal of Field Robotics*, 32(3):315–335, 2015.

Own Publications

Journal Papers

Alberto Romay, Stefan Kohlbrecher, and Oskar von Stryk. An object template approach to manipulation for humanoid avatar robots for rescue tasks. *KI - Künstliche Intelligenz*, 30(3):279–287, 2016.

Alberto Romay, Spyros Maniatopoulos, Stefan Kohlbrecher, Alexander Stumpf, Oskar von Stryk, Hadas Kress-Gazit, Philipp Schillinger, and David C. Conner. Collaborative Autonomy between High-level Behaviors and Human Operators for Remote Manipulation Tasks using Different Humanoid Robots. *Journal of Field Robotics*, 2016.

Stefan Kohlbrecher, Alexander Stumpf, Alberto Romay, Philipp Schillinger, Oskar von Stryk, and David C. Conner. A Comprehensive Software Framework for Complex Locomotion and Manipulation Tasks Applicable to Different Types of Humanoid Robots. *Frontiers in Robotics and AI*, 3:31, 2016.

Stefan Kohlbrecher, Alberto Romay, Alexander Stumpf, Anant Gupta, Oskar von Stryk, Felipe Bacim, Doug A. Bowman, Alex Goins, Ravi Balasubramanian, and David C. Conner. Human-robot Teaming for Rescue Missions: Team ViGIR's Approach to the 2013 DARPA Robotics Challenge Trials. *Journal of Field Robotics*, 32(3):352–377, 2015.

Conference Papers

Alberto Romay, Stefan Kohlbrecher, David C. Conner, and Oskar von Stryk. Achieving Versatile Manipulation Tasks with Unknown Objects by Supervised Humanoid Robots based on Object Templates. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 249–255, Nov 2015.

Alberto Romay, Stefan Kohlbrecher, David C. Conner, Alexander Stumpf, and Oskar von Stryk. Template-based Manipulation in Unstructured Environments for Supervised Semi-autonomous Humanoid Robots. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 979–986, Nov 2014.

Alberto Romay, Stefan Kohlbrecher, Alexander Stumpf, Oskar von Stryk, Felipe Bacim, Doug A. Bowman, Alex Goins, Ravi Balasubramanian, and David C. Conner. Hose task at the 2013 DARPA Robotics Challenge trials: Team ViGIR's results video. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 1095–1095, Nov 2014.

Stefan Kohlbrecher, David C. Conner, Alberto Romay, Felipe Bacim, Doug A. Bowman, and Oskar von Stryk. Overview of Team ViGIR's approach to the Virtual Robotics Challenge. In *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*, pages 1–2. IEEE, 2013.

Open-source Software

Romay, Alberto. ViGIR Object Template Library (OTL). https://github.com/team-vigir/vigir_templates, 2015. Accessed: 2015-12-09.

Romay, Alberto. ViGIR Object Template Manager (OTM). https://github.com/team-vigir/vigir_object_template_manager, 2015. Accessed: 2015-12-09.

Wissenschaftlicher Werdegang[1]

07/2004	Preparatory School mit Schwerpunkt in Matematik und Physik an der La Salle University, Mexico
08/2004 - 12/2008	Studium der Cybernetics Ingenieur an der La Salle University, Mexico
12/2011	Master of Science mit Schwerpunkt Cybernetics
seit 10/2012	Doktorand am Fachbereich Informatik, Technische Universität Darmstadt
10/2012 - 03/2016	DAAD Stipendiat
seit 08/2013	Wissenschaftliche Hilfskraft mit Abschluss, Fachbereich Informatik, Technische Universität Darmstadt
10/2013 - 09/2014	SEP Stipendiat

Erklärung[2]

Hiermit erkläre ich, dass ich die vorliegende Arbeit, mit Ausnahme der ausdrücklich genannten Hilfsmittel, selbständig verfasst habe.

1 gemäß § 20 Abs. 3 der Promotionsordnung der TU Darmstadt
2 gemäß § 9 Abs. 1 der Promotionsordnung der TU Darmstadt